ZX COMPUTING MONTHLY

An Argus Specialist Publication
OCTOBER 1986 £1.50

FOR ALL SINCLAIR USERS



SPECTRUM GRANDMASTERS
Colossus 4 V
Psi Chess

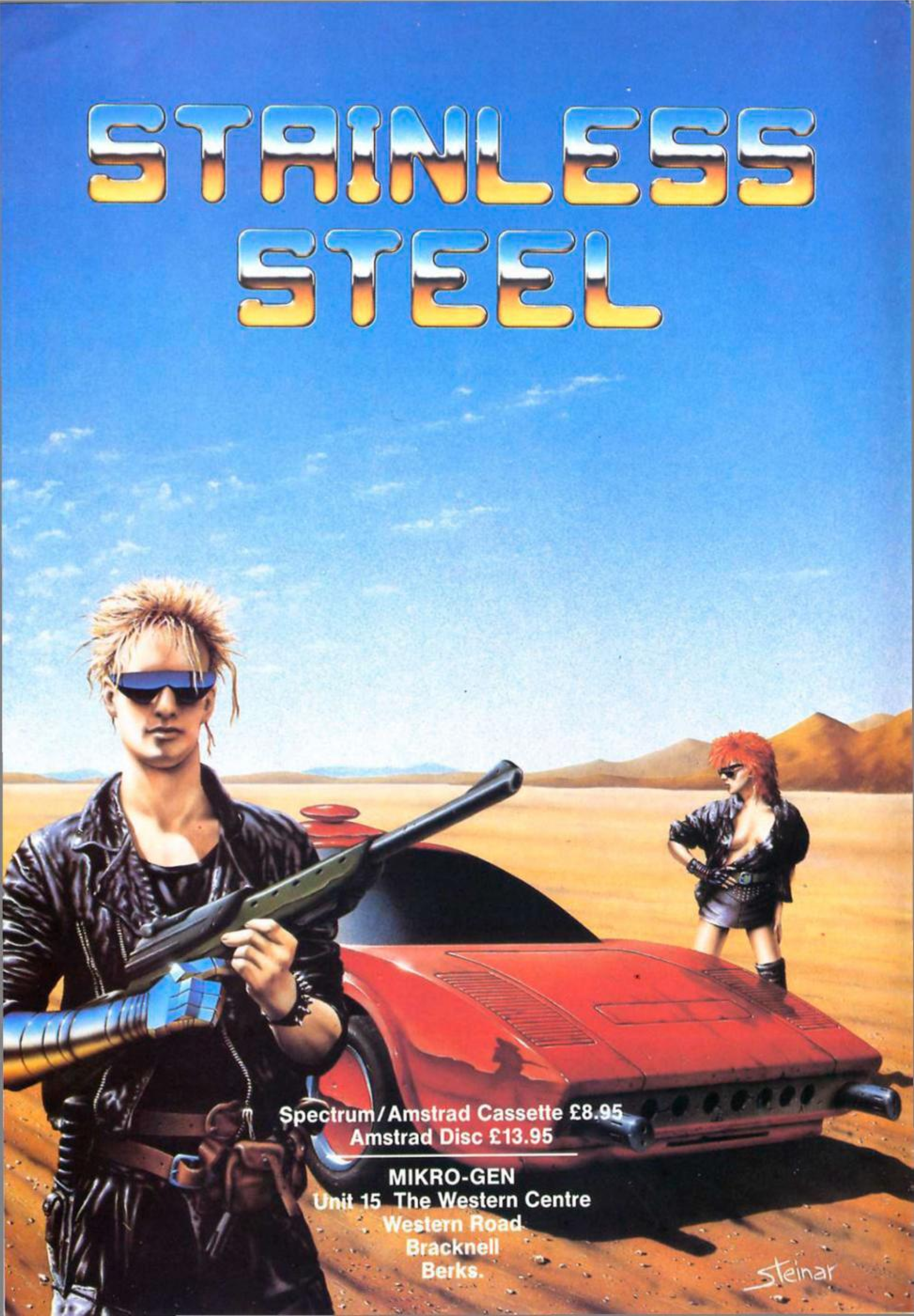


SPECTRUM 128 PLUS 2
Reviewed inside

BUG FREE PROGRAMMING
by Hewson's Steve Turner

REVIEWS ▸ ▸ ▸ ▸ ▸
Dynamite Dan II •
Stainless Steel • Mindstone

WIN! ▸ ▸ ▸ ▸ ▸ ▸ ▸ ▸
Dan Dare • Colossus 4 Chess
Magnum Joystick •


STAINLESS STEEL
Spectrum/Amstrad Cassette £8.95
Amstrad Disc £13.95
MIKRO-GEN
Unit 15 The Western Centre
Western Road
Bracknell
Berks.
Steinar

CONTENTS

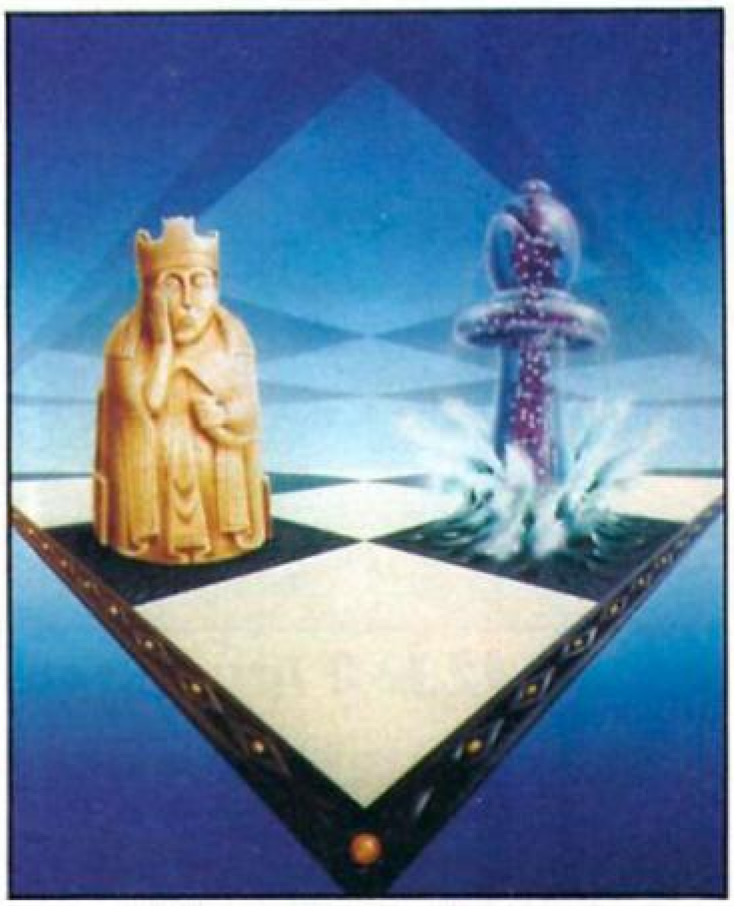
FEATURES

REGULARS



Can you name this game? (41).

COMPETITIONS



Psi Chess from the Edge. Chess feature begins on page 43.

SOFTWARE REVIEWS

PROGRAMMING PROJECTS

NEWS



Spectrum 128+2 launched at PCW Show

Amstrad have made their first move as new guardians of the Sinclair name — an impressive redesign of the Spectrum 128 incorporating features that Spectrum owners have been calling for ever since its introduction.

The 'black box' look of previous Spectrums has gone, replaced by a more robust looking grey/green casing reminiscent of the Amstrad CPC series.

The 'glued on tape recorder' promised by Alan Sugar has been included in the form of an integral datacorder so that fiddling with volume controls on cassette players is now a thing of the past.

The flat keyboard, bane of many a Spectrum programmer has disappeared, to be replaced by a far more professional looking 58 key full travel type writer style 'Qwerty keyboard'.

In keeping with Amstrad's vision of the 128+2 as a powerful game playing machine, two joystick ports have been included. These are non-standard ports accepting only new Sinclair SJS1 joysticks. This is a matching colour coordinated joystick which will be available either bundled with the machine or separately at £14.99. Joysticks that are not compatible with the port can still be attached via interface in the time honoured fashion.

Despite the emphasis on games playing there are still many options open for other uses with ports provided for keypad, serial printer and MIDI. The Plus 2 is compatible with Interface one and microdrives and other peripherals.

So those Spectrum owners who are looking for a computer with more possibilites than games playing will not be disappointed with the capabilities of the Plus 2.

Internally there are no real changes from the original 128, the redesign concentrates on improving ease of use. One minor change in the ROM is however apparent from the title screen, for Sinclair 1982 now reads Amstrad.

One Sinclair tradition that has been adhered to is pricing. The 128+2 comes with a power supply, ariel lead and manual for £149 (inc VAT). That's £30 less than the cost of the mark 1 128 when it was launched.

One quibble with the first 128 was the discovery that some 48K software would not run on it. Amstrad hope to improve this situation by badging Spectrum software with a "Sinclair Quality Control" seal of approval which guarantees that the software will be compatible with the 128+2.

Amstrad have shrewdly redesigned the 128 in the Amstrad image and with a decent keyboard and built in tape recorder now added and a highly competetive price tag it seems that Amstrad under the guise of Sinclair have another success story on their hands.

Getting technical

Spectrum 128+2
£149 (inc VAT)

Memory
RAM 128 Kbytes
ROM 32 Kbytes

CPU
Z80 running at 3.54690 Mz

Screen
256 x 192 Pixel resolution
24 x 32 colour resolution
8 colours foreground, 8 colours background plus intensity and flash settings
Independently controllable border

Sound
Three voice tone and/or noise with 16 envelope setting. One voice CPU generated. Output via TV or audio socket

Keyboard
58 keys full travel 'Qwerty' keyboard

Storage
Integral datacoder. Volatile silicon disc

Firmware
48K Spectrum Basic (compatability mode)
128K enhanced Spectrum basic with full screen editor and men driven command system
Calculator
Single-key program loading system

Interfaces
UPH PAL TV port
Serial printer port
RGB monitor port
Keypad port
Midi port (out only)
2 joystick ports
Audio Out port
Expansion 10 port (full Z80 bus)

Editor: **Bryan Ralph**
Assistant Editor: **Cliff Joseph**
Consultant Editor: **Ray Elder**
Advertising Manager: **John McGarry**

Design: **Argus Design**
A.S.P. Advertising and Editorial
No. 1 Golden Square, London W1R 3AB 01-437-0626

Printed by Alabaster Passmore and Sons Ltd, Tovil, Maidstone, Kent.
Advertisement Copy Controller: Lynn Collis.
Distributed by: Argus Press Sales and Distribution Ltd, 12-18 Paul Street, London EC2A 4JS

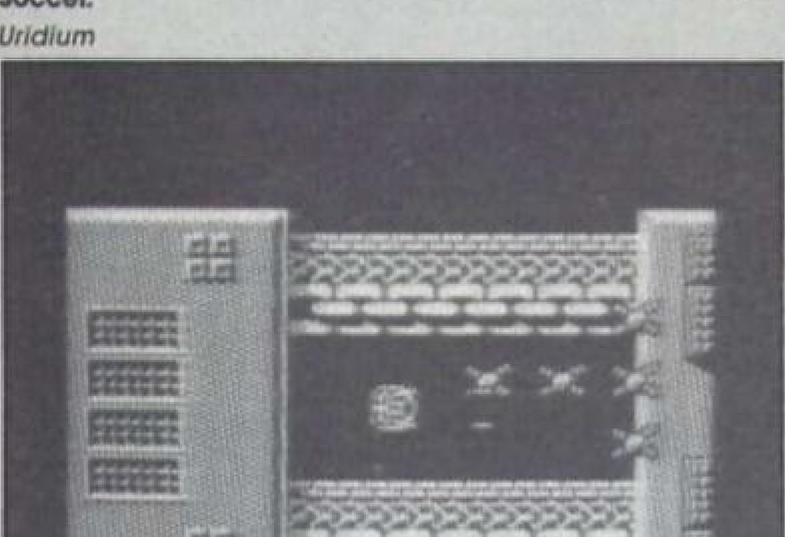
ZX Computing Monthly is published on the fourth Friday of each month. Subscription rates cn be obtained from ZX Subscriptions, Infonet, Times House, 179 The Marlowes, Hemel Hempstead, Herts HP51 1BB.

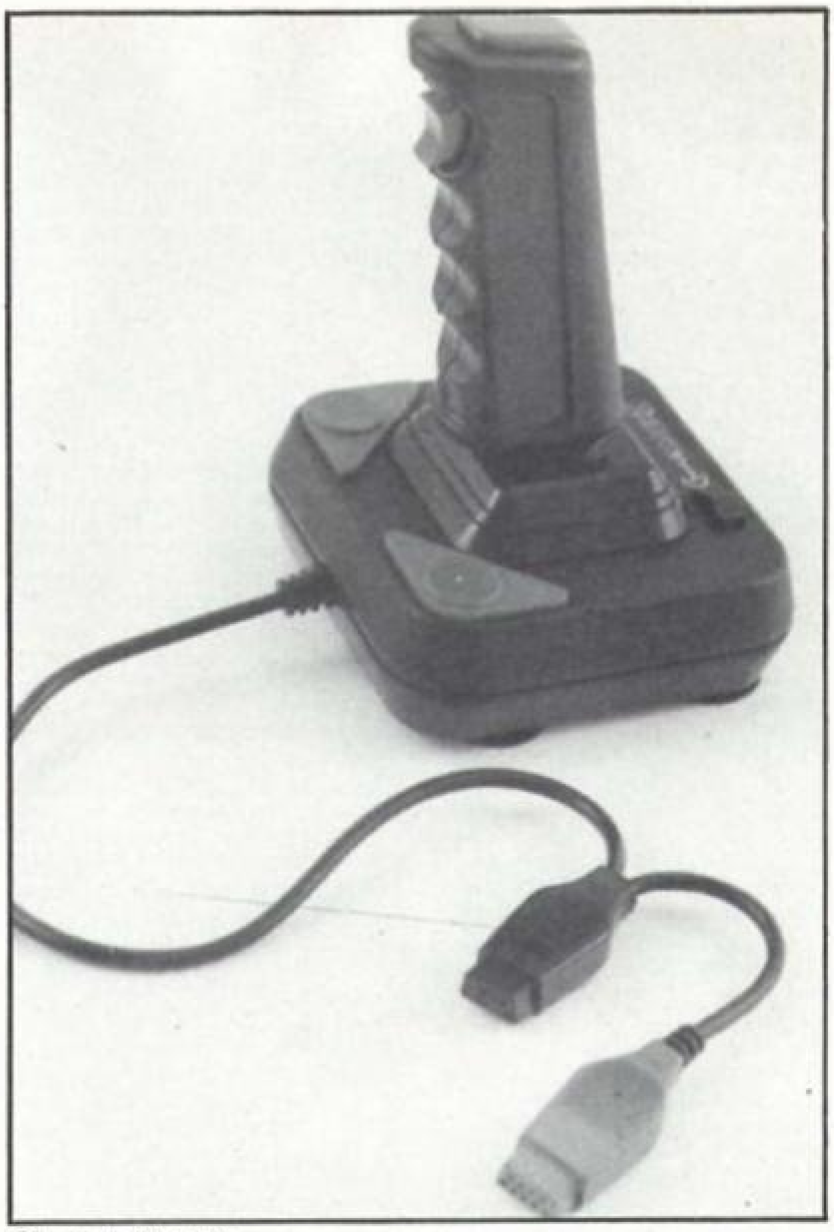
PCW Show Diary

The software houses in previous years have tried to outdo each other by stands of sheer size. This year the status standard had shifted towards height with several of the major companies opting to go for two-tier stands. From their lofty positions software house execs could peer down into the crowded hall below where the serious business of hunting down the latest games was going on. But, as always, a vast majority of what was on view was in an "unfinished" preview state. But here's a taste of things to come in the near (or distant) future from the software houses.

Activision — Spin off games from the **Howard the Duck** cartoon strip and the forthcoming David Bowie film **Labyrinthe.**
Addictive — President, a new game from Kevin "Football Manager" Toms.
Ariolasoft — Deactivators an android bomb disposal team fight against the clock in a 3D maze of rooms. Hanna Barbara spin offs — **Centurion** and **The Challenge of The Gobots.**
Beyond — With a stand decked out like the bridge of the Enterprise Beyond were heralding Star Trek (ETA mid October).
Domark — A new Bond game, **Live and Let Die.**
The Edge — Fairlight II, a full 128K version designed with the Plus 2 in mind. It will load in two parts for the 48K machine. **Shao-Lins road**, an arcade conversion from an original Konami martial combat game. **Palitron** — 3D sci-fi game from the author of Strangeloop.
Electric Dreams — Sci-fi horror film **Aliens** will get the computer game treatment; ETA November. **Dauntless**, an arcade conversion by the Ram Jam Corporation; ETA November. **Explorer**, an arcade adventure set on a 'jungle planet'; ETA October. **Chameleon**, a game based on the battle between the four elements, 400 scrolling screens and 3D'Pana mation! (whatever that is); ETA unknown. **Big Trouble in Little China**, a film spin off with mystical martial arts elements; ETA November. **Circus of Fear**, described as a mixture of "adventure, strategy and arcade" — a murder mystery set under the big top. Other games in the pipeline are **Prodigy** and **Star Raiders II.**
Firebird/Odin — Hypa ball a futuristic sports game and **The Plot** an arcade shoot-em-up.
Hewson — Uridium, a great success on the C64 is being converted. A high speed space combat game which incorporates impressive aerobatic flying. **Firelord**, from the author of Starquake and Devils Lair, an arcade adventure, set in a medieval world.
Melbourne House — Hot on the heels of the Spectrum 128+2 have three books due for release, **Spectrum Plus Two Machine Code for the Absolute Beginner, Spectrum Plus Two Programmer's Guide** and **Spectrum Plus Two ROM Disassembly.** On the games front there is **Asterix the Gaul, Dodgy Geezers** (from the writers of Hampstead and Terromelinos) and **Inspector Gadget.**
Ocean/Imagine — The Great Escape, a 3D POW escape drama, **Parallax**, a space shoot-em-up and film spin offs, **Top Gun** and **Cobra** are just a few of Oceans forthcoming releases. From Imagine there's **Konami's Golf**, a coin op conversion, **Mag Max** and a new soccer simulation — **Super Soccer.**

Uridium





Cheetah Mach1+

New Peripherals at PCW

Cheetah spotted the need for joystick adaptors for the new Spectrum 128+2 and the Cheetah Mach1+ (£14.95) and the 125+ (£8.95) come with standard and 128+2 compatible connectors. An adaptor for standard to 128+2 joystick compatibility is available for £2.99.

Ram Electronics have launched the Music Machine, a complete add-on music system for £49.95. Also released is Ramprint (£34.95), a centronics printer interface with joystick port and 'instant access' word processing.

Spectrum Games Top Ten

1	(1)	Jack the Nipper	Gremlin
2	(2)	Ghosts n Goblins	Elite
3	( )	Atlantic Challenger Game	Virgin
4	(4)	Green Beret	Imagine
5	( )	The Boggit	CRL
6	( )	Mission Omega	Mindgames
7	( )	Stainless Steel	Mikrogen
8	(9)	Bobby Bearing	The Edge
9	( )	Trivial Pursuit	Domark
10	( )	Nexor	Design Design

(Chart supplied by W.H. Smith)

Little change at the top this month as the software houses hold back their new games for en masse release at the PCW show. Jack the Nipper remains at No 1 and the longest stayer Green Beret seems to have permanently claimed the No 4 spot with its third showing just inside the top five.

Bounces Winners

The Bounces competition in our June issue brought in lots of entries (far too many in fact, since I've now got to spend all morning typing out the names of the winners), though quite a few took liberties with the wordsquare and started inventing all sorts of odd words to get the full number required (have you ever heard of a sport called *Polo?*).

Anyway the final list of winners is as follows: first prize of £100 worth of sports equipment — Brian Birch, Ireland.

Ten copies of the Bounces game go out to: Peter Bibby, Cheshire; Gary Bond, County Antrim; Simon Wright, N.Yorks; Kevin Standing, Maidstone; John Luck, Hertfordshire; Mrs R. Mackie, County Antrim; Bruce Allman, Cheshire; Michael C. Payne, Gloucester; Philip Main, Peterborough; David Schwitzer, London.

Fifty Beyond T-shirts for: David Hylden, Wellingborough; K. Watson, Aldershot; Alan Wood, Isle of Wight; Nicholas Whitney, Brewood; Gary Evans, Dyfed; M. Watson, Darwen; D.M. Shannon, County Antrim; Michael Hunderthark, Eastbourne; Chris Geggus, Essex; H. Lyndon, Farnborough; Marcus McManus, County Leitrim; J.S. Russell, Fareham; Peter Taswell, Devon; M. Leston, Blackburn; David Harrow, Hertfordshire; Alistair May, Moray; D.E. Nicholls, London; David Nunes, Ohio, USA; Mrs C. Welsh, Glasgow; R. Sharp, Dundee; Norris McLean, Ayrshire; Gavin Hunt, Nottingham; David Leak, Essex; T. Scragg, Yorkshire; Colin McChesney, Renfrew; Darren Way, Plymouth, L.R. Andrews, London; Don Smith, Sussex; John Alden, Cornwall; K. Devey, Saudi Arabia; A. Rathbone, Warwickshire; Pirotte Eddy, Belgium; N. Coburn, Lancashire; P. White, Blackburn; Paul Beedon, Kent; Paul Horridge, Lancashire; A. Warwick, Tyne and Wear; Neil Fenwick, Middlesex; Miss E.M. Dron, Surrey; Rob Ramshaw, Tyne and Wear; Andrew Moss, Newbury; Mark Syder, Merseyside; I. Ackerley, Derby; R. Sands, Skipton; T.S. Cox, Widmer End; Jason Killgallon; Glasgow; Paul Matthews, Avon; B. Morris, Rochdale; David Kelley, Stevenage; Adam Mayer, Liss.

And vast quantities of Bounces posters are on their way to: Simon Bean, Oxfordshire; Keith Moore, Kent; Steven Smith, Bristol; A. Taylor, Nottinghamshire; Lee Harris, Guernsey; R. Hardman, HMS Sealion; Daryl Jones, Cardiff; Michael Short, Banffshire; Graeme Pettit, Suffolk; Tony Evans, Newcastle upon Tyne; S. Curtis, London; John Dixon, Moray; K.E. Hankin, West Midlands; Guy Davies, Surrey; Sean Alverson, Essex; Michael Elsy, Tyne and Wear; Lawrence Millea, Liverpool; Philip Yeung, Liverpool; Pierloot Francis, Belgium; Matthew Austin, Buckinghamshire; J.P. Smith, South Yorkshire; Matthew Arnold, Buckinghamshire; Gerry Galloway, Liverpool; Clinton Stallard, Dorset; Stewart Catley, Surrey; A.F. Molin, Leeds; Malcolm D. Mackenzie, Lancashire; Neil Owen, Prestwood; Jason Sheppard, Hampshire; Peter Davies, Lancaster; Gary Swindlehurst; John Pullen, London; Papapanagiotu Evangelos, Greece; Joago Snook, Devon; Keith Giscombe, Worcester; Marcus Beer, Kent; Fergus McCabe, Sterlingshire; J.E. Redfern, London; Graeme Copping, Yorkshire; Sean Rowe, Worthing; Mr K. Acheson, Hertfordshire; C.A. Stamford, North Yorkshire; Michael Edwards, Middlesex; Raphael McCreadie, Maidenhead; Mark Summers, Northumberland; R.L. Laidlaw, Birmingham; Uthir Yasin, West Yorkshire; Jeremy Johnson, Burford; Roy Salmons, Gran Canaria; Mark Edwards, County Antrim; K. Galloway, Basingstoke; Andrew Chester, Rugby; Martin Alexis, London; Neil Dyer, Dorset; Colin Bayes, Worcester; Jorgen Jacobsen, Denmark; Hugh McAudey, Belfast; Mrs L.E. Higgins, Nottingham; Miss C. Chipchase, oxford; M. Cunningham-Brown, Warwickshire; D. Piper, Surrey; Vincent Teo, Hertford; A.I. Dalli, London; Neil Peskett, West Sussex; Alistair McLeod, Scotland; Craig Roy, Cumbria; Barry Carter, London; Douglas Porrit, West Sussex; Alan Back, Derbyshire; Paul Robinson, Maryland, USA; A. Thomson, Essex; Leif Kornstaedt, France; Alan Thorne, Cleveland; Timothy Hoodless, Bedford; Martin Dolphin, Preston; Joe Kasa, Hampshire; Jann Romer, Switzerland; Mark Henderson, Glasgow; Philip Chard, Bristol; Robert Fogarty, Eire; Stefan Elsinga, Netherlands; Eric Ferguson, USA; Clyde Soesan, London; M. Bedford-White, Birmingham.

By appointment

8th Day Software, formerly purveyors of cheapie adventures to not as many masses as they would have liked, have gone upmarket. Their next offering costs a full £6.95 and features a group of characters known in the press as The Royal Family. With all your favourites, royal rogues and a plot that doesn't bear repeating the game, simply entitled HRH should be in the shops round about now, or you can contact 8th Day on 051-677-1581.

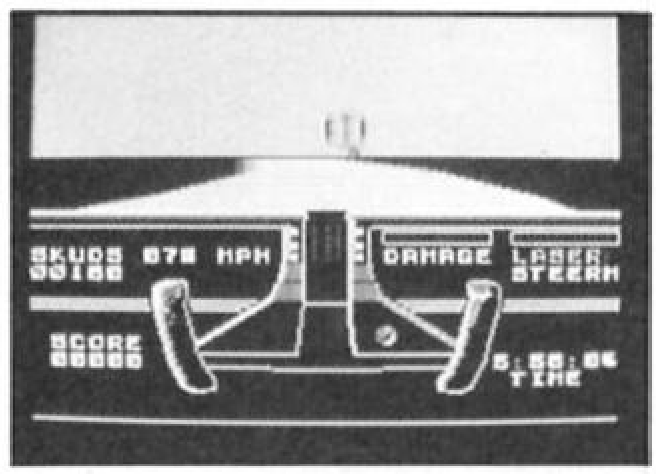


Activision Eidolon

Eidolon is the latest Spectrum offering from Activision, and is due to be released in two versions for the 48 and 128K machines.

The Eidolon is a machine invented to allow you to explore the depths of your own subconscious Id (whatever happened to good old 'zap the aliens' games?). It seems that your mind is full of weird creatures that 'lurk' in the depths, but fortunately the Eidolon can zap these little psychoboogers with fireballs, though it doesn't say what they'll do to the inside of your head.

Featuring 'ultra-realistic' fractual graphics, Eidolon will let you plumb your own depths for £7.99 (48K) or £9.99 (128K).



Knight Rider

Knight Rider—here at last?

By now you might know whether or not Ocean have managed to get Knight Rider out in time to meet its latest scheduled release date which was the end of August. And, as a mark of their confidence Ocean have even supplied some screen shots from the game (though we haven't yet seen a review copy) so you can have a look at it here and see if you think it was worth the wait.

Also on the way from Ocean this September is Nightmare Rally, a "surreal" driving simulation that doesn't look amazingly unlike Knight Rider, and will sell for the same price, at £7.95.



Nightmare Rally

NEWS

Going Ape

Martech are stepping up their release schedule with a number of new titles due for the autumn. The Spectrum version of W.A.R. is due in September, to be followed in October by Cosmic Shock Absorber and a new martial arts simulation called Uchi Mata based on Judo.

Then in November, Tarzan will be swinging along in a game in which the guy with the monkey has to rescue Jane from the natives who have kidnapped her and hidden her in the depths of the jungle (sounds like it's going to offend the sex and racial equality lobbies, but then there's nothing wrong with a little controversial publicity every now and then).

All Martech's autumn releases are priced at £8.95, apart from W.A.R. at £7.95.

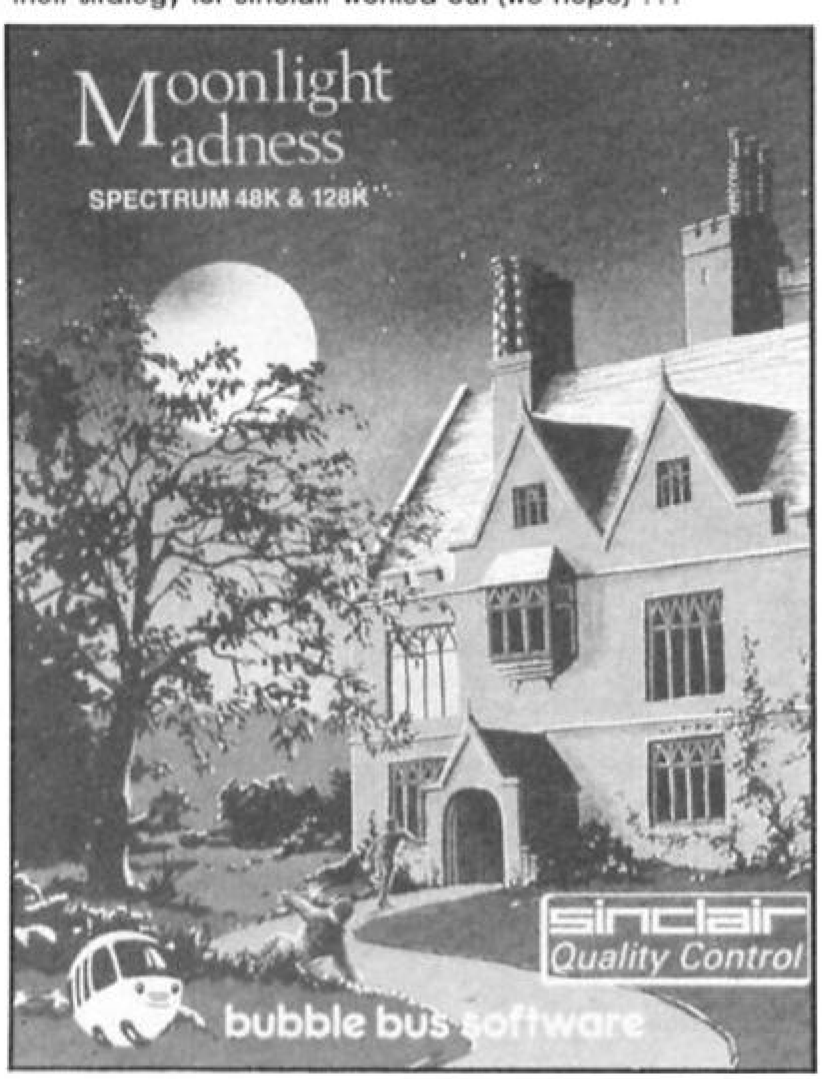
More Chess

Callously disregarding our special Chess feature this month Alligata have decided to launch their new Cyrus II Chess program next month. Available in version for the 48K machines and a special 3D version for the 128, Cyrus II is trumpeted as 'the best Chess program for the Sinclair micros'. Featuring a variety of options (which we'll look at at a later date) both versions of the game will be available in September for £11.95.

Madness by Moonlight

Arriving just microseconds too late to be reviewed this issue, Moonlight Madness arrived from Bubble Bus Software. Written by John Cain, author of Booty — the first ever decent budget game — this is an arcade adventure in which you have to seek out the nineteen pills hidden in a dark old mansion.

We've reproduced the cassette artwork a) because it's got a nice picture on it, and b) because it's the first game we've seen that has the new Amstrad/Sinclair quality control sticker. What with the release of the new Spectrum 128 and the quality control getting started it looks as if Amstrad have finally gotten their strategy for Sinclair worked out (we hope) . . .





Who wants to live forever?

If you do, then Mirrorsoft's new game, Zythum is the game for you. If you can make it across the four territories full of pits, marshes, quicksand, landmines and the like then you just might have a chance of gaining the mystical elixir which will grant you eternal life (which may be the only way you'll ever get to see Knight Rider released). Of course eternal life doesn't come for nothing so you'll have to stump up £7.95 for the privilege.

Citizen Printer Winner

The winner of the Citizen LSP-10 printer is E.B. Hepburn from Dalgethy Bay in Fife, Scotland who correctly answered the questions on the LSP10's print speed in NLQ mode (25CPS), the size of the printer's buffer (4K) and the life expectancy (100,000,000 characters).

Biggles Winners

Lashings of wizard prangs, old chaps and chocks away type banter were received in our Biggles caption contest. But the winner of the Biggles Goodies bag of a copy of the game, Biggles T-shirt, scarf, soundtrack album, badge and poster is A. Hulmes of Altringham in Cheshire for Biggles comment to the threatening German; "Okay, okay, three free flying lessons". Twenty five valiant runners up carry off a copy of the game, T-shirt and badge.

They are Geof Dagger, Blackburn; Chris Harris, Sutton Coldfield; N. Owen, Prestwood; Tony Raven, Llanelli; Andrew James, Tyne and Wear; Charles James, Telford; J. Clayton, Altringham; BJ Kamphuis, Holland; P. Newnham, Weston Super Mare; P. Serbert, Harrowgate; Barry Carter, London E2; Jorgen Jacobsen, Denmark; C. Parkin, Rayleigh; Joseph Kasa, Basingstoke; Gerry Galloway, Liverpool; Christopher Smith, Bushey Heath; Lars Stkholm, Denmark; S. Brodig, Hull; Gary Bond, Carrickfergus; Rob Ramshaw, Tynemouth; Alistair May, Moray; Geof Hill, Belfast; P. Page, Wirral; Ian Hope, Walsall; Christoph Pulhall, West Germany.

Branching out . . .

. . . with two new labels are Ariolasoft. Until recently they've been mainly reliant upon American C64 games for their releases over here, but now Ariolasoft intend to expand with two new labels. Reaktor, which will concentrate on arcade and action games, gets under way with Deactivators. The 39 Steps label will focus on adventure and strategy games, starting with They Stole A Million in which you have to plan the perfect crime. Both new titles will cost £8.95 for the Spectrum version.

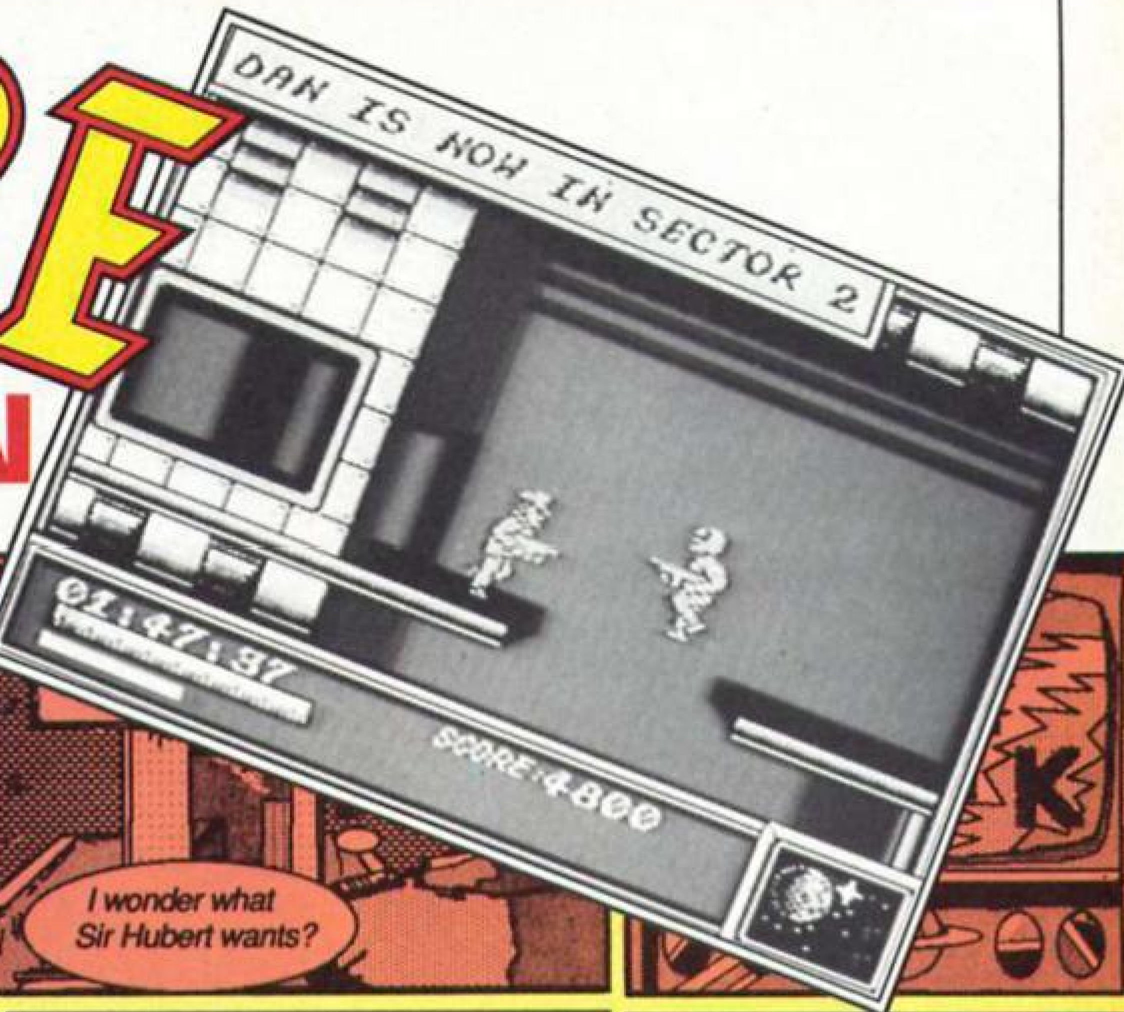
Room 10

CRL look like they'll be having a busy autumn starting with the September release of Room 10. Sounding a bit like a mixture of Ballblazer and squash, Glyding is a futuristic sport played in Room 10 of the Galcorp Leisure Complex, and incorporates a special split-screen display.

Then in October you can expect Cyborg, the first part of a trilogy which carries on the theme of the Tau Ceti game, and this will be followed by the arrival of Dracula. This was originally planned as a simple text adventure, but according to a CRL spokeswoman the game "has become quite sophisticated, and now has 3D graphics".

Dracula and Room 10 will be available for £7.95 while Cyborg will cost £8.95.

DARE TO WIN



Dan Dare is about to burst onto the Spectrum in a new game from Virgin. Win the game and save the universe.

Virgin's Dan Dare is bound to gain accolades from all quarters when it's released soon and we'll be reviewing it fully in the next issue. To whet your appetite it's enough to say that the graphics are among the best yet achieved on the Spectrum and it's a game of non stop action.

The first two winners carry off the game plus a special bonus copy of the book, "The Man Who Drew Tomorrow", the life story of the illustrator Frank Hampson who gave Dan Dare his original look in the comic strip. Packed with illustrations from the classic strips the book retails in the shops for £9.95.

There are a further 25 copies of Dan Dare for runners up.

Quiz time

All you have to do to enter is answer three simple questions on Dan Dare.

1. What comic did Dan Dare originally appear in?
2. What is the name of Dan's arch enemy?
3. What is Dan's spaceship called?

When you've got the answers, complete the entry coupon and send it to Dan Dare Competition, ZX Computing Monthly, No 1 Golden Square, London W1 3AB.

Remember to write your answers on the back of your entry envelope.

The competition is open to all ZX readers from planet earth except employees of Argus Specialist Publications, Alabaster Passmore and Virgin games. The editor's decision is final and no interstellar communications can be entered into. The closing date is November 7th 1986.



Dan Dare Competition

1. ....

2. ....

3. ....

Name ....

Address ....

....

Send your entry to Dan Dare Competition, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB.

COMPETITION


GREMLIN
OF THEIR TIME
Future Knight Trailblazer
C16 + 4 MSX Spectrum 48K
£6.95 £7.95
CBM 64/128
Amstrad
£9.95
Disk
Amstrad Disk
CBM 64/128
Disk
£14.95
Future Knight
Arise Sir Randolph to take up the challenge of death that has befallen the fair maid Amelia, held in the evil clutches of Spegbot the Terrible. Acting upon an inter-dimensional distress signal from the galactic cruiser SS Rustbucket, don your Omnibot Mark IV attack suit and venture forth in pursuit of your beloved's captors. Defend yourself nobly against the Besterka Security droids. Fight your way through 20 gruelling levels onto the planet's surface into Spegbot's castle where the fate of Amelia lies with the outcome of your mortal combat with the awesome Henchodroid. Is there any gallantry and bravery left in this modern day universe?
Available November
FUTURE KNIGHT
TRAILBLAZER
nder
+ 4
.95
November
the Tiger
+ 4
.95
November
a Spring
+ 4
.95
December
Software Ltd.,
Sheffield S1 4FS. Tel: (0742) 753423
Thunder into the unknown at a breakneck speed, pushing your reflexes to their limits in this definitely exhilarating journey that's not one for the fainthearted. Roll left, roll right avoiding the endless chasms of doom that lay in and around the squares of mystery. Squares that will sometimes slow your progress, on occasion with fatal consequences and sometimes speed up unexpectedly or make you jump automatically. Keep a keen eye on the clock as the quicker you complete your task the higher will be your bonus. CBM 64 version is an amazing 2 player simultaneous game. Amstrad Disc version contains extra features.
Trailblazer
Available November

TT RACER

George Du Valle goes for a spin on Digital Integration's new motorcycle arcade/simulation.

T.T. Racer
Digital Integration
£9.95



Was it worth the wait? That question flashed through everyone's head as they had their first look at a playable version of TT Racer, Digital Integration's newest simulation. Although every simulation addict will be familiar with the classic Tomahawk, TT Racer promised to be the first game from DI that could be played, and enjoyed, by anyone.

It took less than ten minutes to get an answer to that initial question, a resounding 'yes'. As the author, R.J. Smith, demonstrated the game, it became very apparent that the two years it had taken to write was well worth it.

Although called TT Racer, the game has very little to do with the actual Isle of Man TT races. In fact, the game is set completely in a Grand Prix environment, with the player being able to race any of the four full classes, 80cc, 125cc, 250cc and the blue riband event, the 500cc.

Options

Even before playing the game, the options give a clue to the incredible scope of this program. In total there are ten initial options, varying from entering your name to the number of additional bikes that will be in the race. Confused? Don't be, DI have managed to make it possible to network up to eight Spectrums together to enable racing. When used (and yes, I have seen it work), it allows you to see whoever is ahead or behind, and they can see you.

Although this is a fabulous option, and one that it would be ridiculous to ignore, especially if you have a few friends with Spectrums, the main game is impressive in its own right. In one load, the game gives you access to any of the 12 world championship tracks, including Silverstone. Then the choices set up your standard: are you a club racer who needs practice for five laps, or an experienced professional who is entering a full GP? The choice is yours.

As with many recently released games, TT Racer has a tie-in, in this case with Suzuki, the Japanese motorcycle producers. However, unlike many tie-ins, Suzuki played a major role in the production of this game. When I first saw the game, it was being played by a small Australian character, who looked totally captivated. It was later I found out that this was Suzuki's Number 1 GP rider, Paul Lewis, who was racing his real life partner, Niall MacKenzie.

What came as the biggest shock was how natural they seemed to be. I have no doubt that Daley Thompson would be hopeless at computer Decathlon, but lap after lap, Paul managed to lower his times. Eventually managing to get his time four seconds faster than the author. Although at the Silverstone GP he unfortunately crashed on the first lap!

Decisions

Even when you have made all the initial options, you must then make a variety of vital decisions, such as tyres. Only then are you ready to start racing. Anyone who has ridden a motorcycle will be at a distinct advantage, even if it is only for a couple of minutes. Getting the knack of accelerating up to nearly 13,000 revs before dropping the clutch is something not many car drivers will ever experience! Having mastered wheelies, and managed to learn the one down, five up, motorcycle gearbox, the game starts in earnest.

Having played the game, I recommend practice on any track you intend to race on. To avoid anything except humiliation, it is necessary to know what kind of corner is coming. Without doubt the most realistic parts of TT Racer are the corners. As with any flight simulator worth its salt, the whole screen leans, and if you are going too fast, you can almost feel the back end sliding away.

Crash

Crashing too is very realistic, to the extent that, as the bike hits the deck, grass is sprayed onto your shattered fairing. Thankfully, however, you manage to avoid Barry Sheene-like broken legs. If, by some miracle, you manage to complete a lap, your time is recorded for posterity, and if it is good enough, it is stored among a roll of heroes such as 'King' Kenny Roberts and 'Fast' Freddie Spencer. (Who? Ed.).

Overall, TT Racer is one of the most impressive games that I have seen on the Spectrum for a long, long time. Not only does it stretch the Speccy to its limits, but it manages to remain incredibly playable. For Digital Integration, it is the most arcade-esque game they have ever produced — and in my view the best.

CROSSFIRE

Readers to the rescue this month with advice and practical help for fellow Sinclair owners

Copying on the Brother

The letter from Janet Watson about COPYing to the BROTHER M1009 printer that you published in the August issue prompts me to send you a listing of my program to do just that.

The program makes use of the bit image mode of the M1009. The problem, of course, is that the pixels on the screen run horizontally whilst bit image mode prints vertically.

What the machine code program does is to take first the eight bytes of the top left character square and store them temporarily. Then it takes the left-most bit of each byte to form a byte corresponding to the first pixel in each of the first eight pixel rows. This is stored as the first byte of A$. Then the second bit of each of the eight bytes are brought together to form the second byte of A$. The process is continued until all 256 pixels of the first eight pixel rows are in A$, then a return is made to BASIC where A$ is LPRINTed in bit image mode. Then everything is repeated for the remaining 23 lines of the screen.

Lines 9980 to 9988 of the BASIC program are intended to be incorporated as a sub-routine within the user's own program. The only stipulation is that A$ must be the first variable in the program.

The result is not quite an accurate copy of the screen because the proportions are different. The printed version is stretched out a bit; it won't produce a proper circle.

I am sorry about the quotes around all the comments in the machine code listing. I wrote the assembler myself and have only just altered it to get the comments printed on the same line as the instruction. I haven't yet worked out how to omit the quotes!

George Brewster.

```
   1 REM copyprnt B 1.1.86
  10 CLEAR 65299: LOAD *"m":1:"copyprnt C"CODE : DIM a$(256)
  20 STOP
9980 OPEN #3:"b": LPRINT CHR$ 27+"3"+CHR$ 24;CHR$ 27+"1"+CHR$ 15
:    REM Open stream to channel "b", set line spacing to 24/216"
     and left margin to 15
9981 POKE 23717,0: POKE 23718,64:
     REM Current screen address = start of display file
9982 FOR i=1 TO 24:
     REM 24 lines on screen
9983   RANDOMIZE USR 65300:
       REM Copy one line of screen to A$ in required order
9984   LPRINT CHR$ 27+"K"+CHR$ 0+CHR$ 1;:
       REM Bit image mode for 256 pixels
9985   LPRINT a$:
       REM Copy one line of screen to the printer
9986   LPRINT CHR$ 10:
       REM line feed
9987 NEXT i
9988 LPRINT CHR$ 27+"2":
     REM Restore normal line spacing
9989 CLOSE #3: RETURN
9998 SAVE *"m":1:"copyprnt B" LINE 1: VERIFY *"m":1:"copyprnt B"
: STOP
9999 SAVE *"m":1:"copyprnt C"CODE 65300,66: VERIFY *"m":1:"copyp
rnt C"CODE
```

Basic Program

Machine Code Program

```
              0000 "1. 1.86 copyprnt T2"
FF14          0005      ORG  65300
FF14          0010 KSAD EQU  5CA5H        : "Current screen address"
FF14          0020 TEMP EQU  5CA7H        : "TEMPorary buffer"
              0025 :
FF14 FD367620 0030 BEGN LD   (IY+76H),20H : "Counter for 32 bytes in line"
FF18 DD2A4B5C 0040      LD   IX,(5C4BH)   : "VARS"
FF1C 110600   0050      LD   DE,0006
FF1F DD19     0060      ADD  IX,DE        : "Point IX to first element of A$"
              0065 :
FF21 2A455C   0070 LP0  LD   HL,(KSAD)    : "HL -> current screen address"
FF24 11A75C   0080      LD   DE,TEMP      : "DE -> start of TEMP buffer"
FF27 0608     0090      LD   B,08         : "Eight bytes to a character square"
              0095 :
FF29 7E       0100 LP1  LD   A,(HL)       : ")"
FF2A 12       0110      LD   (DE),A       : ")Copy the 8 bytes of a character"
FF2B 24       0120      INC  H            : ")square into the TEMP buffer"
FF2C 13       0130      INC  DE           : ")"
FF2D 10FA     0140      DJNZ LP1          : ")"
              0145 :
FF2F 2A455C   0150      LD   HL,(KSAD)    : "Get old screen address"
FF32 2C       0160      INC  L            : "Point HL to next screen address"
FF33 2004     0170      JR   NZ,04
FF35 7C       0180      LD   A,H          : ")If L = 00 add 08 to H in order to"
FF36 C608     0190      ADD  08           : ")point to next screen zone"
FF38 67       0200      LD   H,A          : ")"
FF39 22A55C   0210      LD   (KSAD),HL    : "Store next screen address"
              0215 :
FF3C 0E08     0220      LD   C,08         : "Eight bits in a byte!"
FF3E 21A75C   0230 LP2  LD   HL,TEMP
FF41 0608     0240      LD   B,08         : "Eight bytes in the TEMP buffer"
FF43 CB16     0250 LP3  RL   (HL)         : ")"
FF45 DDCB0016 0260      RL   (IX+00)      : ")Copy one bit from each of the"
FF49 23       0270      INC  HL           : ")TEMP bytes to one byte of A$"
FF4A 10F7     0280      DJNZ LP3          : ")"
FF4C DD23     0290      INC  IX           : "IX points to next byte in A$"
FF4E 0D       0300      DEC  C            : ")Continue until all eight bits of"
FF4F 20ED     0310      JR   NZ,LP2       : ")each TEMP byte have been copied"
              0315 :
FF51 FD3576   0320      DEC  (IY+76H)     : ")Count down the 32 bytes of each"
FF54 20CB     0330      JR   NZ,LP0       : ")screen line and continue"
FF56 C9       0340      RET               : "When finished return to BASIC"
              0345 :
FF57          0350 END. END
END.  65367 FF57H
LP3   65347 FF43H
LP2   65342 FF3EH
LP1   65321 FF29H
LP0   65313 FF21H
BEGN  65300 FF14H
TEMP  23719 5CA7H
KSAD  23717 5CA5H
FF      255 00FFH
$     24741 60A5H
```

Screen problems

Problem No. 1. I use my Spectrum for "serious" work such as word processing (Tasword II, and now Tasword III), financial/technical calculation, for which Omnicalc is excellent, and ***would*** *use it for electronic circuit diagrams* ***if*** *I could find a graphics program which would allow me to draw to an effective screen area rather larger than the Spectrum 256×192 pixel size, and of course Save, and Print to a full-size printer such as the HR5.*

Melbourne Draw is quite adequate for the drawing function, with the facility to define enough UDGs to meet the requirements, and I have Saved screens which are then re-loaded into Lettahead for adding text using the smaller front type offered by the latter. However, MelDraw only offers the standard screen size, and this is hardly enough for even fairly simple circuits. The HR5 printer will accommodate a print width equivalent to almost two

Spectrum screens.

MelDraw does offer the facility to scroll the screen, but the scrolled-off portion is lost and gone for ever! I have tried drawing one screen, Saving it, and then scrolling it almost off the screen, and using the remaining column and/or row as the "seed" for the next screen. However, as you can imagine, the first screen isn't usually right first time, and the complications of recalling, amending resaving, and rescrolling can be imagined.

I am aware of the AMX Mouse, with which an "art" program comes, which does apparently offer a larger effective screen, but at around £80. It's a lot to pay to fill the missing slot in my program inventory, and apart from the larger screen it has a lot of frills which I just don't require. It appears that the program can only be used with the mouse, but I cannot see that the mouse itself offers the facility to use a larger screen: that is the province of the coding, and of course the available memory. (Unless, of course, the mouse has its own ROM in its "cage"!)

Now, has anyone come across what amount to a Melbourne Draw with the additional facility to a) draw to a larger screen area, b) Save same, and c) provide for copying of the larger area to a printer?

(Maybe, if one doesn't exist, Toni Baker could be prevailed upon to come up with an article on how it can be done.)

Problem No. 2 is using Omnicalc with an Opus Discovery. The latter can use the same commands as the Microdrive for Loading and Saving, and the Microdrive version of Omnicalc can be saved to disc. It will run OK as well, but with one or two snags which can be disastrous! For example, if one selects the "Amend" option, and then "Inserts" or "Deletes" to copy or erase a column or row, the program NEWS itself, and with it the data you have just entered...

As Omnicalc users will know, the program as supplied on tape apparently looks for some indication that a microdrive is present, and if it finds one, converts itself to a microdrive version, and saves itself to the drive. However, loading with an Opus connected does not effect a Save as described, even if the **space** *occupied by the MDRV variables is created by the "Rand USR 4007" command available with the Opus for the purpose. Presumably Omnicalc looks for something other than a* **space** *which would be occupied by code bytes if actually associated with a genuine microdrive.*

I bought the Omnicalc extension tape at the last Microfair from a supplier who said it would overcome the problem: whilst it does offer the opportunity to add some useful commands to Omnicalc, one thing it does not do is solve the Omnicalc-on-Opus problem.

Again, any ideas?

Paul King, Hassocks, Essex.

Opus/Tasword

I read with interest the letter in the August issue from Janet Watson about printers and the Opus Discovery and problems. Most of these problems have been solved by writing to Tasman Software, who have been extremely helpful.

If any of the programs Janet has will allow her to save screen$ to tape, a version of Tascopy on 3.5 inch disk for the Discovery can be purchased from Tasman to enable her to print screen copies provided the printer is "Epson compatible". Full instructions are included with the disk.

Before saving Tasword Two to disk, two additional lines must be included in the Basic program to allow control codes to be sent to the printer. These are:-

6 CLOSE # 3
8 OPEN # 3;"b"

Andrew Bailey, Bradford

Checking Machine Code

I must commiserate with Mr F. Wade Cooper of Burgess Hill on his machine code checking problems. I enclose two unsophisticated programmes which I have been using for some years and which may help our friend.

Having been a subscriber to ZX since late 1982 I must say that I wouldnot miss a copy and look forward to its arrival with great anticipation at the articles and programming hints.

R. Roby, Tasmania, Australia

Many thanks to Mr Roby for his thoughtful contribution, I'm sure that the two programs will be of great use both to Mr Wade Cooper and others who run into problesm with machine code programming.

```
10 POKE 23658,8
20 INPUT "AT WHICH ADDRESS DO YOU WISH TO BEGIN CHECK ?",A
30 CLS
40 FOR I= 1 TO 10
50   PRINT A;"....";PEEK A
60   LET A=A+1
70 NEXT I
80 PRINT AT 15,0;"PRESS M..FOR MORE   C.. FOR CORRECTION   F.. TO FINISH"
90 PAUSE 0
100 IF INKEY$="M" THEN GOTO 30
110 IF INKEY$<>"C" THEN GOTO 250
120 INPUT "WHICH ADDRESS NEEDS CORRECTION ?",C
130 PRINT C;"....";PEEK C
140 INPUT "ENTER CORRECT CODE",D
150 PRINT C;"....";D
160 PRINT "IS THIS CORRECT ?...Y/N"
170 PAUSE 0
180 IF INKEY$="Y" THEN POKE C,D:GOTO 210
190 IF INKEY$<>"N" THEN GOTO 160
200 GOTO 120
210 PRINT "ANY FURTHER CORRECTIONS THIS PAGE ?":PAUSE 0
220 IF INKEY$="Y" THEN GOTO 120
230 IF INKEY$<>"N" THEN GOTO 210
240 GOTO 30
250 IF INKEY$<>"F" THEN GOTO 80
260 CLS:PRINT AT 10,0;FLASH 1;"RE=SAVE YOUR CODE NOW"
270 STOP
```

▲ Program 1 Check M/C Program — used to check and alter an existing M/C program

▼ Program 2 used for entering machine code with or without checksum facility and iwth input correction

```
10 POKE 23658,8
20 INPUT "ADDRESS OF FIRST ENTRY ?",A
30 INPUT "NUMBER OF ENTRIES BEFORE CHECKSUM ?(Enter 12 if no checksum available
",N
40 CLS:LET T=0
50 FOR I=1 TO N
60   INPUT "ENTER CODE (Enter 999 to finish)",C
70   IF C<0 OR (C>255 AND C<>999) THEN GOTO 60
80   IF C=999 THEN GOTO 210
90   POKE A,C
100   PRINT A;"....";C
110  LET A=A+1:LET T=T+C
120 NEXT I
130 INPUT "CHECKSUM (Enter zero if no checksum)",S
140 IF S=0 THEN GOTO 240
150 PRINT "CHECKSUM= ";S,"TOTAL= ";T
160 PAUSE 200
170 IF S<>T THEN GOTO 220
180 PRINT "ANY FURTHER ENTRIES ?":PAUSE 0
190 IF INKEY$="Y" THEN GOTO 40
200 IF INKEY$<>"N" THEN GOTO 180
210 PRINT FLASH 1;"SAVE YOUR CODE NOW":STOP
220 PRINT "THERE IS AN ERROR  .PLEASE CHECK"
230 PAUSE 200
240 PRINT "DO YOU WISH TO MAKE ANY CORRECTIONS ?":PAUSE 0
250 IF INKEY$="N" THEN GOTO 180
260 IF INKEY$<>"Y" THEN GOTO 240
270 INPUT "WHICH ADDRESS NEEDS CORRECTION ?",P
280 INPUT "ENTER CORRECT CODE",Q
290 LET T=T-PEEK P+Q
300 POKE P,Q
310 PRINT P;"....";Q
320 IF S<>0 THEN PRINT "CHECKSUM= ";S,"TOTAL= ";T
330 GOTO 240
```

Pen Pals

I'm a 17 year old Spectrum owner from Israel and I would be delighted to swap ideas, pokes, programs and games with anyone from anywhere in the world. I have both a 48 and 128K Spectrum.

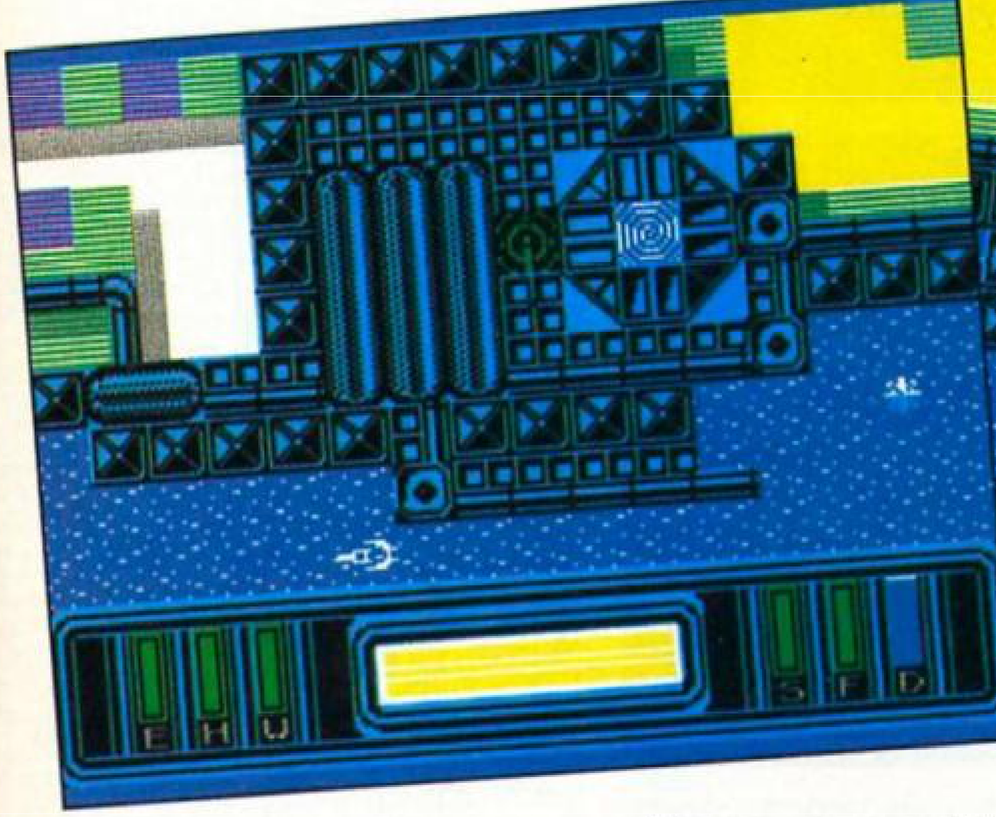
Guy Windreich, Sderot Hatates 5, Ramat-Gan 52453.

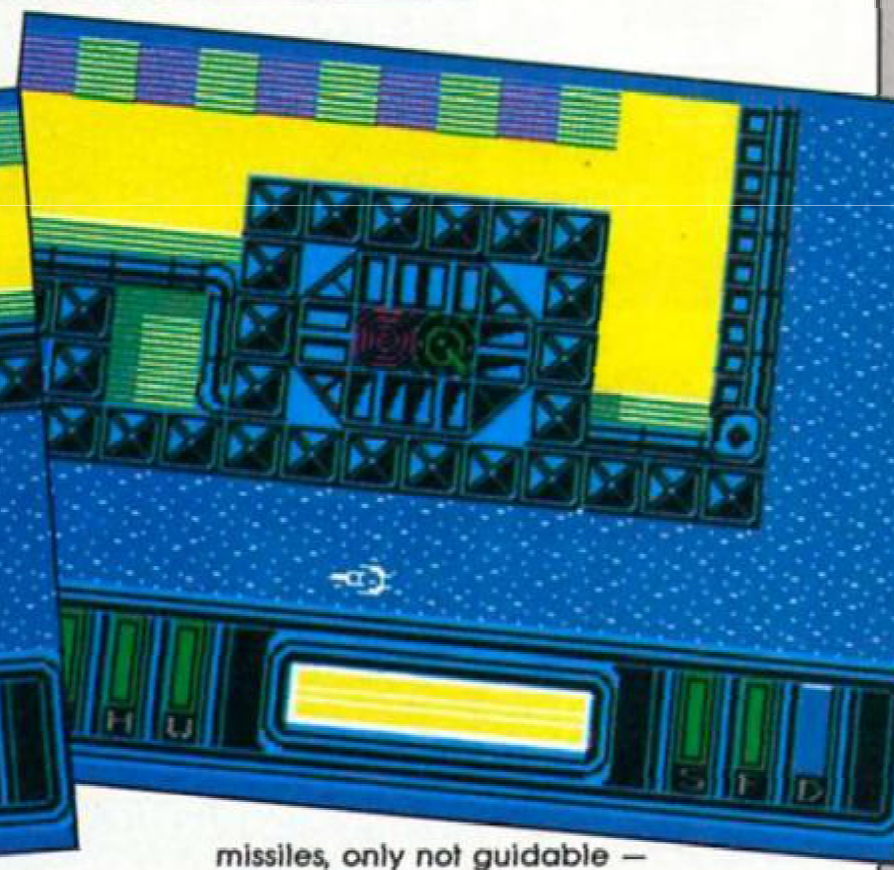
I'm 16 years old and would love to have a pen pal. I have a 48K Spectrum but unfortunately games, books and program listings are very scarse here.

Wilhelm Offermeier, PO Box 20052 Noordbrug, 2525 South Africa.

Skim across the lakes of Xargon in your hydroboat, turn the highways to waterways and destroy the central computer.

XARQ





XARQ
Electric Dreams
£9.95

Reputations are funny things. Some software houses establish themselves with their first game, and then build on this with a constant flow of excellent games. Some houses produce a whole heap of junk, and Electric Dreams are somewhere in the middle. Being one of the top ten software houses in the U.K., they have produced such classics as Spin Dizzy, but they have also produced some less than brilliant games.

Many people will, at first glance, write Xarq off as nothing but an over complicated shoot-'em-up, but with perseverance, and not a little manual dexterity, you can find a good game in there.

In a world where there is no land, a computer was programmed to create a mass on which the inhabitants could live, this world was called Xargon. Everything went well until one day the computer developed a fault, and started to destroy the land it had taken so long to create.

As an average run of the mill superhero, it is your task to enter Xargon and flood the Zimmerman trenches, destroy the Central power reactor, and thereby disable the computer. Pretty simple, huh? Nope! Unfortunately, all of Xargon's defences are fully operational, and the computer is determined not to let you get in!

To help you on your mission you have been equipped with a Nik-Nik Hi-speed Hydroboat, and a great deal of firepower. For some surprising reason, the United Planets' Co-operative has every faith in you.

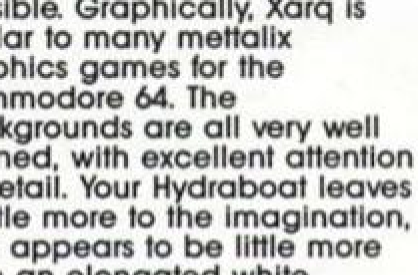
Using either a joystick (Kempston or Interface 2), or keyboard (especially defined for those with nineteen fingers), you must struggle against fighters, gun turrets, and a variety of other annoyingly mean weapon bases.

To help you in your task, you have a variety of different weapons. As well as a standard laser, there are guided missiles, mortars, and depth charges. Guided missiles are especially useful if you want to fire inland, as they are the only weapons capable of knocking out the large and deadly missile turrets. Mortars are the same as guided missiles, only not guidable — surprising eh!

Depth charges are essential if you are to gain more than 5%, as submarines have a tendency of sneaking up unnoticed. On screen there are also a series of displays and scanners that give a continually moving report on how much damage your vehicle has sustained.

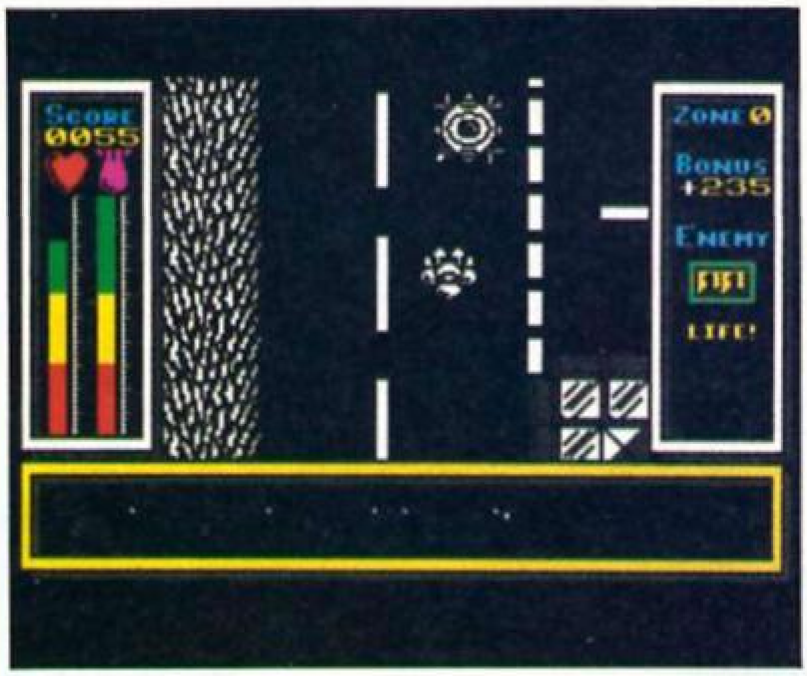
In Xarq, the trenches are arranged concentrically, around the reactor, and it is therefore essential to get as far in as possible. Graphically, Xarq is similar to many mettalix graphics games for the Commodore 64. The backgrounds are all very well defined, with excellent attention to detail. Your Hydraboat leaves a little more to the imagination, as it appears to be little more than an elongated white splodge!

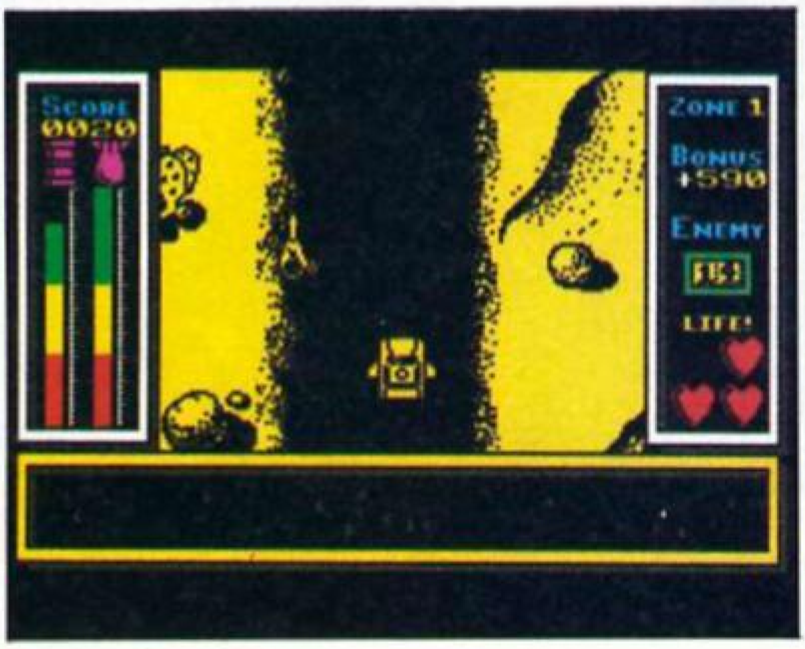
Xarq may not be the best game to come from Electric Dreams, and it certainly isn't the easiest, however, if you are willing to spend a few hours getting used to the control, and working out what to do, you will be well rewarded.

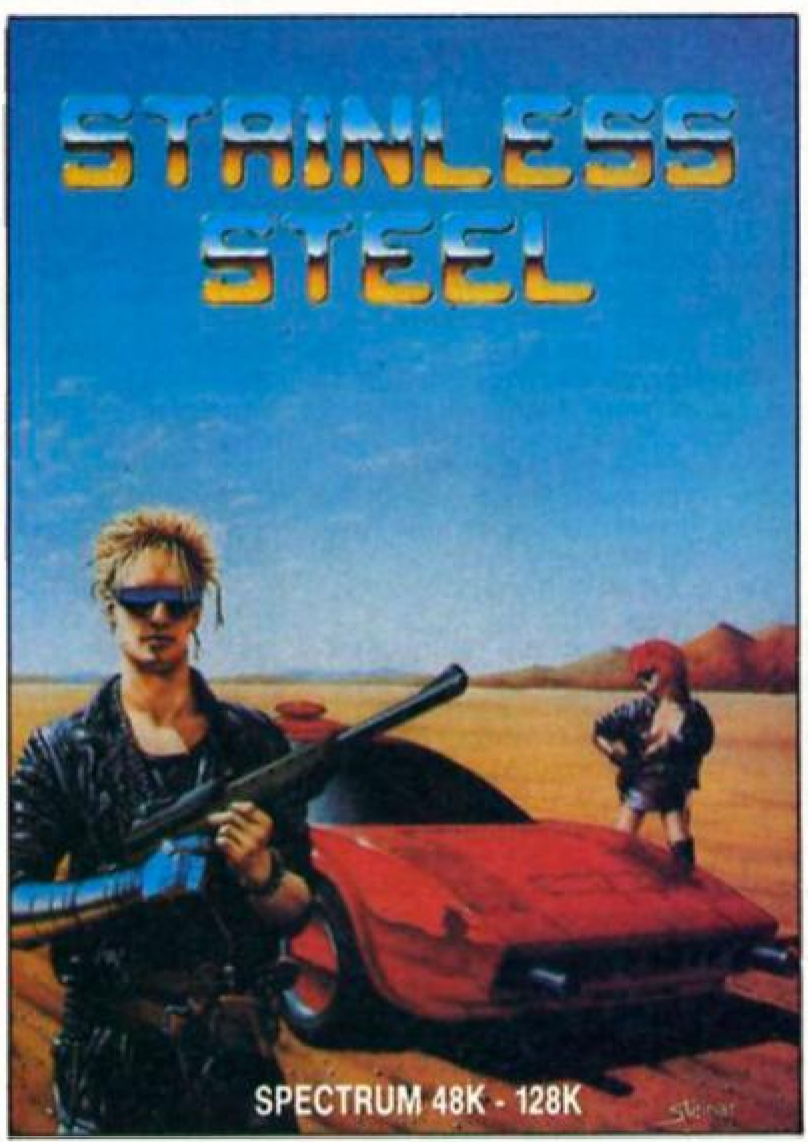


Stainless Steel
Mikro-gen
£8.95

STAINLESS STEEL







Steel yourself for a classy shoot-em up from Mikro-gen.

This is a game to test the mettle of hardened shoot-em up fans. You play "teenage superhero" Ricky Steel who is spending his formative years blasting everything in sight, but it's all in a good cause as the excessively evil Dr Vardos is intent on conquering earth and Ricky is out to thwart him.

There are four phases to fight through before the ultimate showdown with the Doc. In zone one ricky is a pedestrian in search of his "hyper alloy combat vehicle" Nightwind. He sets off up-screen along a smooth scrolling road swarming with android troops, enemy helicopters etc etc. The main screen gives an overhead view of the devastating proceedings and the graphics are large and well defined. Below the animated action is a road map indicating the position of approaching enemies and one eye has to be kept on this as there is little time to react if you are unprepared especially if you are attacked from behind.

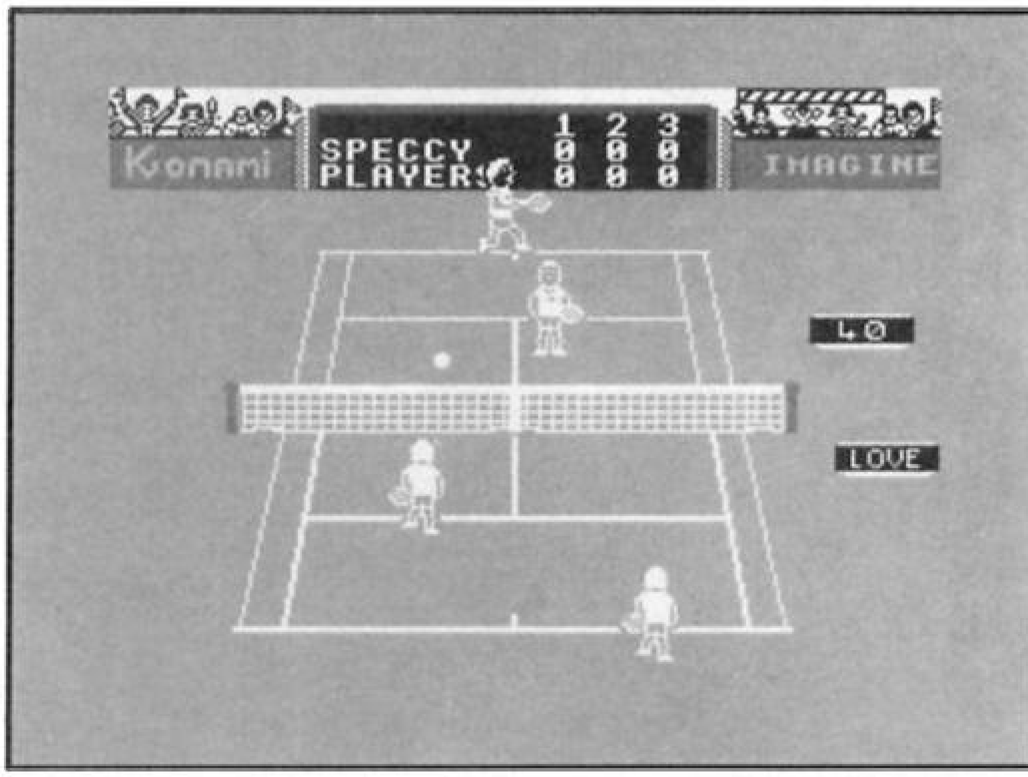
You have four lives and believe me you need them. Losing a life is spectacular — Ricky totally disintegrates with fragments scattered in every direction accompanied by a rather unpleasant scream from the Spectrum beep box.

In the next zone you team up with the car and wipe out more waves of android troopers and in the final zone you are airborne using "uranium thunderbombs" to blow submarines out of the water (Using these bombs in the previous car-leg merely results in self destruction). The final leg of the game is taken up with a search and destroy mission to get Dr Vardos struck off permanently.

Stainless Steel is a graphically sophisticated shoot-em up with enough sequences to maintain your interest. If you enjoy a sustained bout of blasting this game is worth checking out but be warned it requires an agile wrist and quick reactions in fact if you happen to have a steel arm already make sure your wrist is well oiled.

GREAT

Konami's Tennis
Imagine
£7.95



"And that rather weak second service got exactly what it deserved." And out come all the old Dan Maskell stock phrases, not to mention jokes about John McEnroe, iceborgs, and the like.

Yes, it's tennis conversion time and the old arcade favourite, Konami's Tennis finally lobs its way onto the Spectrum courtesy of Imagine's licensing deal with Konami. To be honest, when this arrived I wasn't exactly seized with raptures of enthusiasm and the general reaction was along the lines of "Ho hum, yet another Imagine arcade conversion." But once the game was loaded up it managed to bring a little jollity to an otherwise cold, wet and dreary morning.

On the two occasions that I'd attempted to play the original arcade game I'd found it completely unplayable but the

Imagine serve up top seed tennis game.



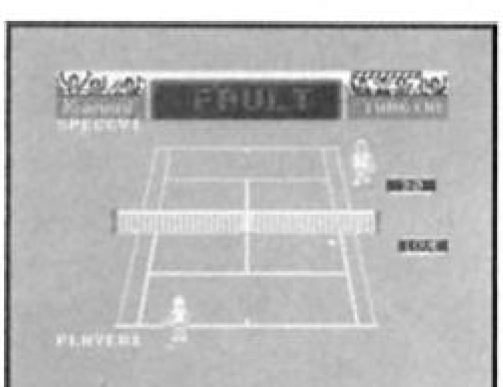
TENNIS

Spectrum version proved to be much easier and more enjoyable to play and although I haven't come anywhere near winning even a single set I can see myself loading this up quite a lot 'till I get the hang of it.

Imagine's conversion is something of a no-frills job, but with the emphasis on playability. There isn't a great range of different shots that you can play with lots of angles, topspin and so on as has been the case with some tennis and table tennis games (which I generally find far too complicated to play well). The direction that you're running in seems to have some effect on the direction that you hit the ball in and the timing of the shot can affect its length, but on the whole it's just a matter of chasing after the ball and thrashing away like mad in the hope that you'll knock it over the net. I realise that that may not sound terribly sophisticated but it's definitely great fun.

The graphics aren't exactly spectacular — the players are white sprites on a green court, with the scoreboard at the rear of the court — but by keeping things simple you can at least see what's going on the whole time, and there are (virtually) no attribute problems.

The options available allow one or two players to play a singles match against each other. Playing doubles against the computer is just asking for trouble. It's chaos, with players running all over the place and the computer managing to co-ordinate its two players with an ease that will make most human opponents look pretty daft.



After a while you'll be able to tighten up your game and maybe even win a few points (the computer does seem vulnerable to a fast serve and volley game). The game seems to have just the one skill level, so it might eventually be possible to master the game and perhaps it would have helped ensure that the game has a long life by providing different skill levels. Other than that my only criticism of the game is that using the keyboard controls is a bit fiddly.

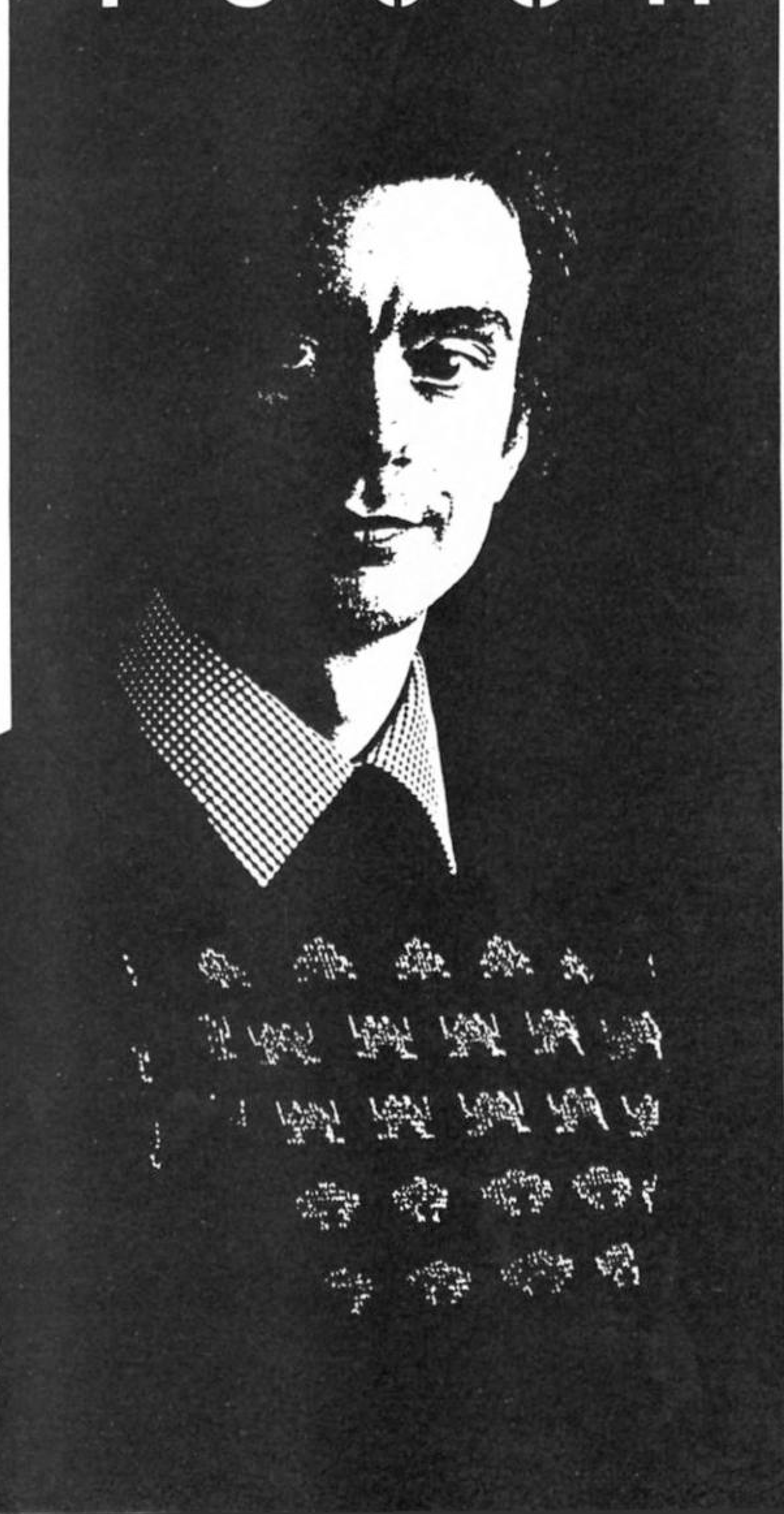
The predefined control keys seem to be badly chosen and although you can redefine these keys player two can't choose keys set aside for player one even if player one is using a joystick. This means that if player two wants to redefine the keyboard controls then he or she will probably also have to redefine player one's keyboard controls first, even though player one isn't using the keyboard (and this isn't explained in the instructions either, it took a bit of guessing and trial and error before I began to realise why I couldn't get the keys I wanted for player two).

Still, fiddly bits aside, Konami Tennis has still proved to be the first tennis game on the Spectrum that I've enjoyed playing and is certainly the only one that I'm ever likely to go back to.



GREAT

THE PROFESSIONAL TOUCH



Everything you always wanted to know about bug-free programming as Steve Turner focuses on program testing.

The software for the USA "Starwars" project will take something in the order of 3,000 man years to write. Many people think that it is impossible to write a vast system without including bugs that would make it a worthless heap of coding. It will have to work properly the first time it is used operationally. There will be no second chance.

Well the destiny of the world is not dependent on our programs but we face the same problems. Why do programs always seem to have bugs? Why do programmers make so many mistakes? Can we avoid making them or at least eliminate them from the finished program?

We all make mistakes. I once heard a nurse state that if she made as many mistakes as a programmer did her patients would not survive.

The fact is she probably makes just as many errors but few are critical despite her work. Most activities that humans perform are not critical. There is a whole area of success with a gradual transition to failure that allows plenty of room for our natural inaccuracy.

When we do slip up because we are constantly monitoring our actions we can spontaneously make a correction. It is only when this constant feedback fails to spot a mistake that things start to go wrong. Even then another person will often be involved for any important action and the mistake can be spotted and corrected.

But when computers are involved it is different. Unable to recognise a silly command, a machine will mindlessly do its best to carry it out regardless of the consequences. We store up all our natural mistakes in a program. This is then blindly executed by the machine without the benefit of our inbuilt monitoring and feedback facilities.

The result of any mistake is remote from when it actually occurred. The time between making a mistake and spotting and correcting it may be hours, days or even years. But the process is essentially the same. We all make mistakes but programmers' mistakes have a delayed action, like a time bomb!

Can we avoid making mistakes? It may be impossible to be perfect but we can try to minimise the mistakes we make. The most important factor is to

really admit to ourselves that whatever our experience we make mistakes. Then we stand a chance of avoiding them or at least finding them later.

The way to avoid errors is by discipline and I find it sometimes is in direct conflict with the creative process. If you can reach a compromise between the urge to write and the control to do it properly you will be a much sought after programmer.

Trapping the bugs

Eliminating bugs is a question of setting traps at every stage of the design and programming process. It is no good finding that during programming the overall design just does not work. There is never time to keep on checking everything so much that all bugs are found. We have to whittle down the number of bugs by looking in the places they are most likely to occur.

Design checks

1. Program size
Estimate the size of each file or table, buffer, map area etc. Guessing the size of the actual program is harder but base it on previous programs that you have written. Write down a list of every routine required as a guide. Check when writing you are coding within the size you have allowed for an average routine.

2. Program speed
Often speed is critical, especially in animated games. Calculate execution times of the critical routines such as refreshing the screen. Try to estimate a frames per second figure for your program. Adjust the screen area and cpu load accordingly.

3. Game design
To check the game concept try a BASIC or paper mock-up. Explain the game to someone and note their ideas etc. Try to think of all possible ways people could misplay the game.

4. Program objectives
Very often, especially in business, the design does not achieve all that is required. First you must decide exactly what you are trying to achieve. Before you start the design make a list of the requirements and use it as a checklist when your design is complete.

5. The human machine interface
If you are trying any new control method try it out first before you commit yourself to it. Is it too difficult? Can the average user handle it?

6. Program structure
Test the overall program structure chart or flowchart by reading it through, to someone else if possible. Explain how it works and invite questions such as "what happens if?". Play the game in your mind following your diagram in execution sequence. Be the machine and try to execute the program drawing pictures and diagrams of what is happening. Write notes as to when the main variables are initialised or updated.

Programming errors

I have drawn up a Top of The Bugs chart listing the machine code errors that are most often made in Z80 assembler. These are well worth looking out for when a routine has been coded before you waste time trying it out. It is always advisable to read your code when you have finished writing it especially if you have keyed it straight in without putting it on paper first.

You can only ever see a very small portion of code on the screen. I am a believer in printing new routines so I can have a quick dry run. I do not go round loops for every iteration but at the very least I try the first and last cases.

For calculations I work through considering the maximum and minimum values writing down the possible range after each step. For bit shifting or bit manipulation I draw bit diagrams to show what happens. Of course many times I cannot be bothered to do this until the computer proves me wrong! It always gets done sooner or later.

The other day I had a large routine that worked first time. I was so dismayed at this unusual event I spent the next two hours pulling the routine apart confident there must be a bug somewhere. . .and there was, a real time bomb just waiting for me to forget all about that bit of code.

Run testing

Despite all precautions some bugs still manage to survive until the routine is run. Testing should be systematic. Test small portions of code at a time. Then if it crashes you know where the error is. Gradually construct the program by adding small untested bits to the reliable bit. If any parts are self contained get them working on their own first then try them in the main program.

I work the same way in BASIC although I will test groups of lines out as I write them checking variable contents out etc.

Play/user testing

This stage is not so much to find bugs but to see if the program is achieving its aims. As other people than the programmer are involved unexpected inputs take place and often more bugs are found. I tune a game up at this stage to optimise its playability etc.

The final trap

When one of my games is ready for launch it undergoes one final

Top Ten Bugs

1. Variable or register not initialised properly.
2. Condition reversed (eg. JR C instead of JR NC).
3. Reversed operand order (eg. LD A,B when LD B,A is required).
4. Data and addresses confused (eg. LD HL,1000 when LD HL,(1000) is required).
5. Loop counters incorrect. Each must be initialised, fetched, updated, stored.
6. The zero case forgotten. What happens on a zero loop count or null input?
7. A register or variable is used for a second purpose and the first value not restored. Very often occurs when subroutines are called or extra lines have been inserted.
8. Arithmetic errors, especially when a mixture of 16 and 8 bit numbers are involved.
9. Communication between routines. Define the inputs and outputs of each routine.
10. Confusing hex and decimal numbers.

How to reduce your mistakes

1. Design before writing.
2. Structure your programs.
3. Document your program.
4. Try to write as little new coding as possible. Use old trusted routines, macros or copy similar bits of coding.
5. Write in a consistent manner. For instance, always decrement loop counters in the same place in the loop.
6. Keep your coding as simple as possible. Do not use clever tricks.
7. Leave the stack alone.
8. Write new routines on paper first.

test. I document the whole code from scratch reading every routine making sure I understand it. The odd last bug is found in this way. Then it is the public's turn.

Making a test plan

To run test a routine properly we need to be systematic. The best way to do this is to make a test plan which can be used as a checklist as each test is complete. It also helps when test data has to be made up.

Each test can be given a reference number. This is written alongside the record of test data that has been made up to test that case. Very often the test data will be human input such as joystick commands.

You can still be systematic and list all the specific conditions you must create. You can use a program listing as your test plan. Mark on it or a separate piece of paper each group of instructions between any conditional instructions, eg. JRz , DJNZ , JP C etc.

A test is needed to make each set of instructions execute. Identify the condition and the function of the chunk of code and you have the test that is required. When it comes to the actual testing many of the tests on your list can be ticked off easily without any specific setting up. You will then be left with specific conditions to set up.

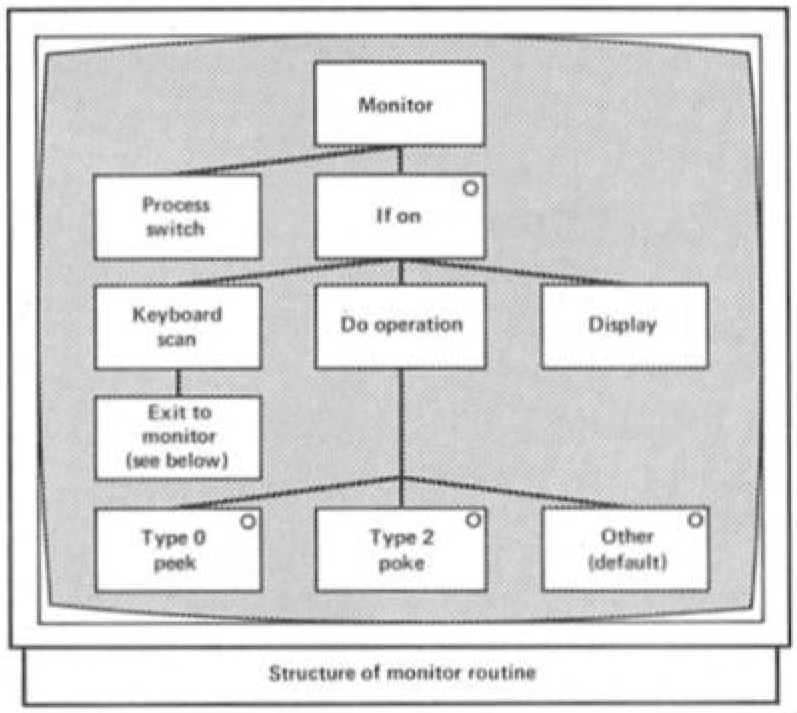
Prove each section of code really did its job. That does not mean just seeing if it runs without crashing. Decide where you will break the routine to check all registers and variables are set up correctly. You do not have to do this for every group of instructions unless it is a really tricky routine. At the end of each subroutine for example, is a good place to do this.

The list of functions to check serves another important purpose. It should include every major task that the design requires of that routine. A routine must not only work, it has to do the right job.

An example Test Plan

Here is a test plan which you can use to check the example MONITR routine works. Before reading the list see how you can do. Turn to the program and identify all the functions to test. Then compare your list to mine below. I have included the name of the routine performing each function.

Monitor Test Plan



Structure of monitor routine

1. MONITR — No action if MONON = "0" if any key but "1" is pressed.
2. MONITR — When "1" is pressed MONON is set to "1".
3. MONITR — If MONON=1 routine is executed whatever key is pressed except "0".
4. DELAY — Is speed slow enough?
5. KEYSCAN — Exit working if "0" is pressed.
6. KEYNO — Are keys "5", "6", "7" and "8" recognised and control passed to each correct task. If none are pressed does nothing happen
7. KEYNO — Right cursor test. Does it increment the MONDIG number to a maximum of 6?
8. LEFTCR — Left cursor test. Does it decrement the MONDIG number to a minimum of 0?
9. DOWN — Causes digit to roll down.
10. UP — Causes digit to roll up.
11. TURNBT — Is each digit pair in turn updated?
12. TURNBT — Do left and right digits in a pair update?
13. OPTYPE — Does MONOP = "0" cause a peek, "2" cause a poke and anything else do nothing.
14. PEEK — Getting memory?
15. POKE — Changing memory?
16. HEX — Digit pairs decoded?
17. NULLOP — Print file built up OK?
18. COLMON — Attribute cursor moves OK? Does it miss

How to find that hidden bug

1. Start looking at the lines you last changed. The bug is usually there.
2. Try to locate the error by executing less of the program. Execute each big routine in turn using a monitor until the error occurs. Then do the same for each section of the faulty routine.
3. Write BASIC programs to validate any large data files.
4. Talk through a problem area with a friend. You usually see the error as you explain it.
5. Sleep on really tough problems. Next morning you often find the answer just comes to you.
6. Print off the problem area, dry run it and plan a search strategy using breakpoints.

How to debug a BASIC program

This list of tips should help you if you have typed in a program from a magazine and it does not work.

1. Look at the magazine listing. If it is typeset there may be a printing error but the error is probably in your copying. If it looks like a printer listing any mistake will definitely be in your typing in.
2. Check problem areas like pokes, data lists etc.
3. Execute the program a bit at a time by inserting STOP statements. Then print any changed variables to see what is happening. Then use CONT to carry on the program.
4. Insert some extra print statements in loops etc to print out loop counters or other variables.
5. Look for all the clues. Write down what exactly happens. Try it again. Vary the input and try to pin down the exact circumstances when the error occurs. Then try to find routines that process that eventuality.

gaps? is the default colour left behind?

19. PUTMON — Is the mini dump picking up the right data and placing it in the print file?

20. HEXTWO — Is the print OK?

Well, that is my list. Note the similarity between the list and the structure diagram. If I had taken the structure to a low enough level one box on the diagram would be one test. If you want you can use a structure diagram as a test plan provided you list all functions within a box.

Although the list may seem long it takes about 30 seconds to run through the tests. That is if they work. If they do not you have a precise definition of the error condition.

Real Time Monitor

This month's example program acts as a window into the machine. Unlike most monitors it is very small and operates as your program executes. You can watch your variables as they change.

You can speed up testing by applying values to variables (as long as somewhere else is not resetting them). This is invaluable for tuning up games. Constants can be changed in flight to get speeds right, positions right, rates of firing adjusted etc.

Merely call the monitor from within your mainline loop, I put it in the keyboard scan routine then it even operates in the game's pause mode. This means by pausing the game you can set up several variables.

The routine includes an exit from your game if the 0 key is pressed. The jump address should be changed to the entry point of your monitor or back to BASIC routine. I use the breakpoint entry point of my main monitor which allows me to get back and continue the test if I want to.

To find the entry address to your monitor breakpoint routine set a breakpoint and look to see what instruction and address has been placed. It may be a call or a jump. Change the line JP EXITAD accordingly.

To use this facility always start off the test session from your main monitor so it can initialise its variables. Then jump to the program under test.

You can call the routine from within any loop you are testing and you will have an escape if it loops forever. There is a delay within the monitor which can be reduced if the rate of call is slow.

The routine prints at the top of the screen using the print routine in the August issue. Alternatively you could use RST 10 instructions by loading b=MONOUT as a count and HL to step along MONLIN.

Instructions for use

1 = switch monitor on
5 = cursor left
8 = cursor right
6 = decrease digit
7 = increase digit

Set op. Type to 0 to PEEK selected address.

This is the same as the first byte of the dump. If the value is changing too fast to read freeze it changing the op type from 0 to 1.

1 to set up selected byte. No update of memory takes place yet (default mode).

2 to POKE selected byte to the selected address. The monitor is dangerous if left in this mode. Change the op type back to zero unless you are block filling.

Listing

```
0000                ;
0000                ;  EXAMPLE DISPLAY
0000                ;
0000                ; 6000 00 1   00010203 04050670
0000                ; ↑
0000                ; address
0000                ;
0000                ;      ↑
0000                ;      selected byte
0000                ;
0000                ;         ↑
0000                ;         op type
0000                ;
0000                ;            ↑
0000                ;            dump of 8 bytes
0000                ;
0000                ;
0000                ;
0000                      ORG  0B000H
B000                MONITR DL   $
B000 3A27B1               LD   A,(MONON) ;ON SWITCH
B003 A7                   AND  A
B004 200F                 JR   NZ,ISON   ;IF NOT ON
B006 01FEF7               LD   BC,0F7FEH ;TEST 1 KY
B009 ED78                 IN   A,(C)
B00B F6FE                 OR   0FEH
B00D FEFF                 CP   0FFH
B00F C8                   RET  Z ;RETURN IF MONON=0
B010 3E01                 LD   A,1   ;IF 1 PRESSED
B012 3227B1               LD   (MONON),A
B015 210040         ISON  LD   HL,4000H ;SLOW IT
B018 2D             DELAY DEC  L        ;DOWN
B019 20FD                 JR   NZ,DELAY
B01B 25                   DEC  H
B01C 20FA                 JR   NZ,DELAY
B01E                ;KEYSCAN
B01E 01FEEF               LD   BC,0EFFEH
B021 ED78                 IN   A,(C)    ;XXX67890
B023 F6E2                 OR   0E2H;SET UNUSED BITS
B025 6F                   LD   L,A
B026 01FEF7               LD   BC,0F7FEH
B029 ED78                 IN   A,(C)    ;XXX54321
B02B 17                   RLA  ;         XX54321X
B02C F6DF                 OR   0DFH     ;..5.....
B02E A5                   AND  L        ;..567890
B02F FEFE                 CP   0FEH
B031 2003                 JR   NZ,NOQUIT
B033 C30FF7               JP   EXITAD ;###CHANGE
B036                ;THIS LINE TO YOUR MAIN MONITOR
B036                ;ENTRY POINT
B036
B036 FEFF           NOQUIT CP  0FFH   ;PROCESS KEY
B038 CA86B0               JP   Z,OPTYPE
B03B 1F                   RRA
B03C 1F                   RRA
B03D 0E00                 LD   C,0
B03F 1F             FINDK RRA  ;FIND FIRST BIT ZERO
B040 3003                 JR   NC,KEYNO
B042 0C                   INC  C
B043 18FA                 JR   FINDK
B045 79             KEYNO LD   A,C        ,UPDATE
B046 2128B1               LD   HL,MONDIG
B049 1129B1               LD   DE,MONADD
B04C FE03                 CP   3
B04E 2811                 JR   Z,LEFTCR
B050 FE02                 CP   2
B052 2816                 JR   Z,DOWN
B054 FE01                 CP   1
B056 2816                 JR   Z,UP
B058                ,RIGHT CURSOR
B058 7E                   LD   A,(HL)
B059 3C                   INC  A
B05A FE07                 CP   7      ,MAX 6TH DIGIT
B05C 3028                 JR   NC,OPTYPE
B05E 77                   LD   (HL),A
B05F 1825                 JR   OPTYPE
B061 7E             LEFTCR LD  A,(HL)  ,CURSOR LEFT
B062 FE00                 CP   0       ;0=MINIMUM
B064 2820                 JR   Z,OPTYPE
B066 3D                   DEC  A
B067 77                   LD   (HL),A
B068 181C                 JR   OPTYPE
B06A 0EFF           DOWN  LD   C,0FFH  ; LOWER
B06C 1802                 JR   TURNBT
B06E 0E01           UP    LD   C,1     ; HIGHER
B070                TURNBT
B070 7E                   LD   A,(HL)  ;=DIGIT NO
B071 47                   LD   B,A
B072 2600                 LD   H,0
B074 CB2F                 SRA  A       ;/2 FOR BYTE
B076 6F                   LD   L,A
B077 19                   ADD  HL,DE
B078 78                   LD   A,B     ,HIGH OR LOW
B079 E601                 AND  01      ,NIBBLE
B07B 79                   LD   A,C
B07C 2006                 JR   NZ,TOMONB
B07E 17                   RLA
B07F 17                   RLA
B080 17                   RLA
B081 17                   RLA
B082 E6F0                 AND  0F0H
B084 86             TOMONB ADD A,(HL)
B085 77                   LD   (HL),A  ;UPDATE
B086
B086                ;DO OPERATION
B086 112EB1         OPTYPE LD  DE,MONLIN
B089 3A29B1               LD   A,(MONADD)
B08C 67                   LD   H,A
B08D 3A2AB1               LD   A,(MONADD+1)
B090 6F                   LD   L,A    ;HL=SELECTED
```

```
B091 3A2CB1    05390           LD   A,(MONOP) ;ADDRESS
B094 A7        05400           AND  A
B095 2806      05410           JR   Z,PEEK
B097 FE20      05420           CP   20H
B099 2809      05430           JR   Z,POKE
B09B 180A      05440           JR   NULLOP
B09D 7E        05450   PEEK    LD   A,(HL)
B09E 322BB1    05460           LD   (MONVAL),A
B0A1 1804      05470           JR   NULLOP
B0A3 3A2BB1    05480   POKE    LD   A,(MONVAL)
B0A6 77        05490           LD   (HL),A
B0A7           05500
B0A7 2129B1    05510   NULLOP  LD   HL,MONADD  ;DISPLAY
B0AA CD07B1    05520           CALL HEX     ;PAIR 1
B0AD 23        05530           INC  HL
B0AE CD07B1    05540           CALL HEX     ;PAIR 2
B0B1 13        05550           INC  DE      ;GAP
B0B2 23        05560           INC  HL
B0B3 CD07B1    05570           CALL HEX     ;PAIR 3
B0B6 13        05580           INC  DE      ;GAP
B0B7 23        05590           INC  HL
B0B8 CD07B1    05600           CALL HEX     ;LAST DIG
B0BB 3E20      05610           LD   A,32    ;BLANK UNWANT
B0BD 3237B1    05620           LD   (BLANK),A
B0C0 3A28B1    05630           LD   A,(MONDIG)
B0C3 210058    05640           LD   HL,5800H
B0C6 0620      05650           LD   B,20H
B0C8 360F      05660   COLMON  LD   (HL),0FH   ;ATTRIB
B0CA 23        05670           INC  HL         ;DEFAULT
B0CB 10FB      05680           DJNZ COLMON
B0CD 210058    05690           LD   HL,5800H
B0D0 FE04      05700           CP   04         ;MISS GAPS
B0D2 3806      05710           JR   C,YE
B0D4 23        05720           INC  HL
B0D5 FE06      05730           CP   06
B0D7 3801      05740           JR   C,YE
B0D9 23        05750           INC  HL
B0DA 1600      05760   YE      LD   D,0
B0DC 5F        05770           LD   E,A
B0DD 19        05780           ADD  HL,DE     ;CURSOR
B0DE 3630      05790           LD   (HL),30H ;COLOUR
B0E0           05800
B0E0 3A29B1    05810   PUTMON  LD   A,(MONADD)
B0E3 67        05820           LD   H,A
B0E4 3A2AB1    05830           LD   A,(MONADD+1)
B0E7 6F        05840           LD   L,A
B0E8 1138B1    05850           LD   DE,MONDMP
B0EB 13        05860           INC  DE         ;MINI HEX
B0EC 0604      05870           LD   B,4        ;DUMP
B0EE CD07B1    05880   HEXDMP  CALL HEX
B0F1 23        05890           INC  HL
B0F2 10FA      05900           DJNZ HEXDMP
B0F4 0604      05910           LD   B,4
B0F6 13        05920           INC  DE
B0F7 CD07B1    05930   HEXTWO  CALL HEX
B0FA 23        05940           INC  HL
B0FB 10FA      05950           DJNZ HEXTWO
B0FD 110000    05960           LD   DE,0000H   ;Y,X POS
B100 212DB1    05970           LD   HL,MONOUT ;FILE
B103 CD4AB1    05980           CALL PRINT
B106 C9        05990           RET
B107           06000   ;HEX TO ASCI CONV
B107           06010   ;HL=ADDRESS OF HEX DIGIT PAIR
B107           06020   ;DE=ADDRESS OF ASCII
B107 AF        06030   HEX     XOR  A ;SPLIT  2 DIGITS
B108 ED6F      06040           RLD
B10A 12        06050           LD   (DE),A
B10B 13        06060           INC  DE
B10C ED6F      06070           RLD
B10E 12        06080           LD   (DE),A
B10F 13        06090           INC  DE
B110 ED6F      06100           RLD  ;SETS (HL) AS IT WAS
B112 1B        06110           DEC  DE
B113 1B        06120           DEC  DE
B114           06130
B114 0E02      06140           LD   C,2     ;ASCII CONV
B116 1A        06150   DIGIT   LD   A,(DE)
B117 FE0A      06160           CP   0AH
B119 3004      06170           JR   NC,DOLET
B11B C630      06180           ADD  30H
B11D 1802      06190           JR   DONE
B11F C637      06200   DOLET   ADD  41H-0AH
B121 12        06210   DONE    LD   (DE),A
B122 13        06220           INC  DE
B123 0D        06230           DEC  C
B124 20F0      06240           JR   NZ,DIGIT
B126 C9        06250           RET
B127 00        06260   MONON   DB   0 ;0=OFF 1=ON
B128 00        06270   MONDIG  DB   0 ;CURSOR DIGIT NO
B129 00AF      06280   MONADD  DW   0AF00H ;SET TO YOUR
B12B           06290   ;                   VARIABLES
B12B 00        06300   MONVAL  DB   0 ;VALUE OF BYTE
B12C 00        06310   MONOP   DB   0 ;OPERATION TYPE
B12D           06320
B12D           06330   ;OUTPUT FILE
B12D 1C        06340   MONOUT  DB   1CH ;NO OF CHARS
B12E 00000000  06350   MONLIN  DB   0,0,0,0,32,0,0,32
B132 20000020
B136 00        06360           DB   0
B137 20        06370   BLANK   DB   32
B138 20000000  06380   MONDMP  DB   32,0,0,0,0,0,0,0,0
B13C 00000000
B140 00
B141 20000000  06390           DB   32,0,0,0,0,0,0,0,0
B145 00000000
B149 00
F70F           06400   EXITAD  EQU  63247;PICTURESQUE
B14A           06410   ;MONITOR ENTRY POINT
```

SHORTCUTS

Ray Elder presents an expanded Short Cuts this month reflecting the programming talent of ZX readers. So if you have an astonishing routine hidden away why not flaunt it in Short Cuts. Remember there's £10 awarded for published routines and the £20 Starcut prize for the outstanding mini-programme of the month.

Sounding Out

Robert Glavas of Bury supplies our first program, a routine to detect sound entering the cassette port. This could form the basis of many programs. He supplies us with a demo showing how it could create graphs.

A couple of points. First, some older computers need to be allowed to warm up for about five or ten minutes before the signal settles to a standard, and secondly you have to provide the sound!

This can be done in several ways, either connect a microphone to the ear socket, but this may not be very successful, or record a word or message and play it back into the Spectrum by the usual lead, or put your recorder into "play" and speak into the built in mic.

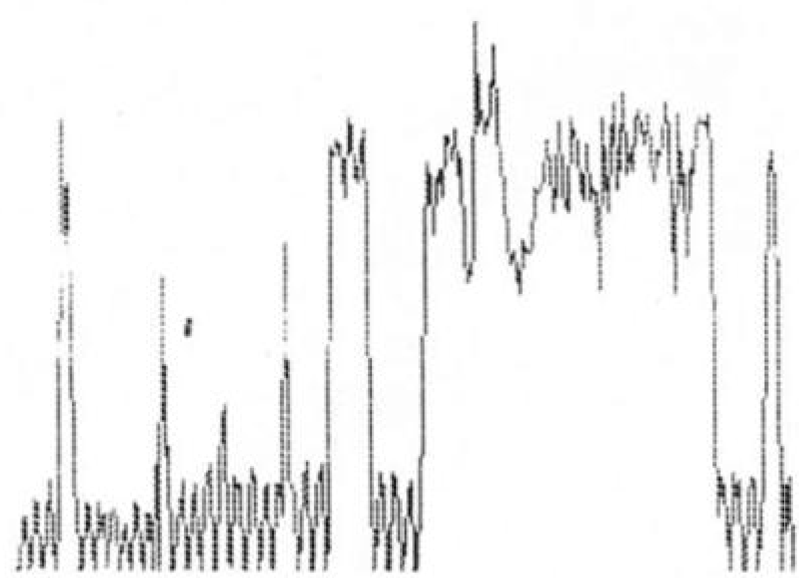
Testing 1, 2, 3, 4. Testing 1, 2, 3...

```
   1 REM sound sensor
  1Ø DATA 33,129,92,54,Ø,6,1,219
,254,2Ø3,119,32,1,52,16,-9,2Ø1
  2Ø FOR f=5ØØØØ TO 5ØØ16: READ
a: POKE f,a: NEXT f
```

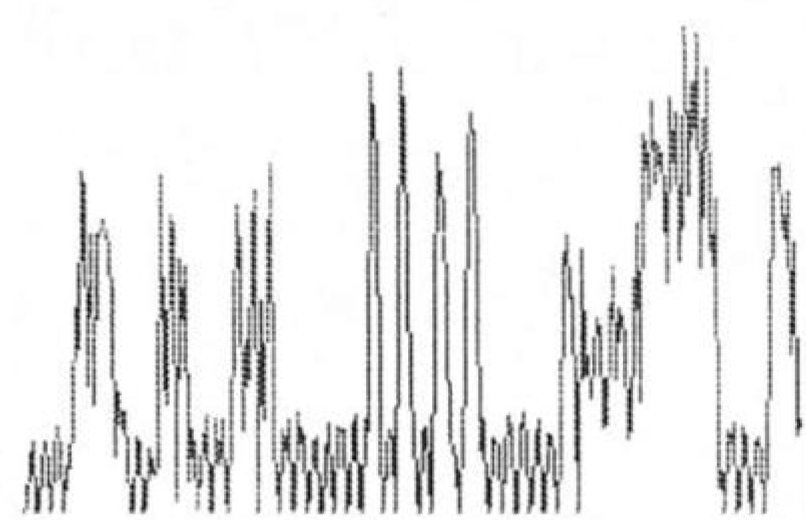
```
 1ØØ REM demo 1, graph generator
 11Ø LET s=1
 12Ø POKE 5ØØØ6,7Ø
 21Ø LET x=s: LET o=Ø
 3ØØ LET l=USR 5ØØØØ
 31Ø LET a=PEEK 23681
 32Ø PLOT x-s,o: DRAW s,a-o
 33Ø LET o=a
 34Ø LET x=x+s: IF x>255 THEN  L
ET x=s: CLS
 35Ø GO TO 3ØØ
```

```
 1ØØ REM demo 2, try whistling !
 11Ø POKE 5ØØØ6,5Ø
 12Ø LET l=USR 5ØØØØ: LET a=PEEK
 23681: IF a>1Ø THEN  BEEP .Ø1,3
Ø
 13Ø GO TO 12Ø
```

HERE'S A SCREEN DUMP:



AND ANOTHER:



Sound kit error

B. J. Kamphuis is a programmer who hails from The Netherlands and is a regular ZXC contributor and he has asked us to make a correction to his last published program.

In the program we published in our July issue the code should have read in the THIRD line 35,126,35..etc and NOT 35,125,35. Slap on the wrists all round, but meanwhile here is another sample of his work, a simple but interesting machine code routine to convert numbers between 0 and 255 to Hexadecimal.

Load the program and RUN it. Lines 9,10 and 20 contain the main M.Code and to use it in any adaption of your own POKE 64000 the number, call 64002 and collect the results as CHR$ from 64000 and 64001. A little exercise would be to make it deal with numbers greater than 255 (I have, for my own use!).

```
   1 REM dec to hex
  10 DATA 58,0,250,245,6,4,203,6
3,16,252,205,28,250,50,0,250,241
,230,15,205,28,250,50,1,250,201,
198,48,254,58,216,198,7,201
  20 FOR f=64002 TO 64035: READ
a: POKE f,a: NEXT f
 100 REM  demo
 110 INPUT "Dec Value ? ";d
 120 POKE 64000,d
 130 RANDOMIZE USR 64002
 160 PRINT d;" Hex Value=";CHR$
PEEK 64000;CHR$ PEEK 64001
 170 GO TO 110
```

Double Up

And now to Tim Rance of Bucks. He sent a short, simple BASIC routine to print a message in double height. I know we have used other variations before, but it's neat and there must be thousands of new readers who are looking for such a routine!

```
   1 REM double print
   2 REM Characters A&B in line
        20 are entered in
        GRAPHICS mode
  10 LET x=0: LET y=0: INPUT "En
ter Message..."; LINE a$
  20 FOR a=1 TO LEN a$: LET c=15
616+(8*(CODE a$(a)-32)): FOR b=0
 TO 7: LET d=(USR "a"+(b*2)): PO
KE d,PEEK (c+b): POKE (d+1),PEEK
 (c+b): NEXT b: PRINT AT y,x;"A"
;AT y+1,x;"B": LET x=x+1: NEXT a
```

VU Calc Controlled

Another reader from Holland, J.K.H. van den Bosch sent us a routine to allow all users of VU CALC to print out onto a proper printer, although there is a built in COPY for Sinclair type printers, only four columns are printed at a time.

To use this program make sure your DATA has been saved then quit the program, NEW the computer and LOAD and run the program given. Lines 1,2 and 3 include interface control codes for the Kempston E Centronics I/F. They perform the following functions and you may have to modify them if you are using a different I/F.

Line 1. turn the Sinclair tokens off.
Line 2. POKE 23679 sets the number of characters per line.
Line 3. Checks, and puts the printer into condensed mode if the number of characters per line is greater than 80. Some printers may print more without needing to be changed, and some printers may use different codes, check your manual.

A useful program, anyone done anything similar for VU FILE?

```
   1 COPY : REM CHR$ 0
   2 CLS : INPUT "Number of colu
mns?(MAx 10)";x: LET x=x*VAL "7"
: POKE VAL "23679",x+SGN PI
   3 IF x>VAL "80" THEN  LPRINT
CHR$ VAL "27";CHR$ VAL "27";CHR$
 VAL "15"
  10 INPUT "Start at Column?";y:
 LET y=y-SGN PI
  20 INPUT "Start at Row?";z$: I
F LEN z$=SGN PI THEN  LET z$=" "
+z$
  30 IF z$(2)="" THEN  LET z=COD
E z$(1)-VAL "97"
  40 IF z$(2)<>"" THEN  LET z=(C
ODE z$(2)-VAL "97")+(VAL "26" AN
D z$(1)="a")+(VAL "52" AND z$(1)
="b")
  50 INPUT "Number of Lines? ";r
: LET r=r-SGN PI
  60 LET q=VAL "30477"+(VAL "7"*
y)+(VAL "420"*z)
  70 FOR f=q TO q+(r*VAL "420")
STEP VAL "420"
  80 FOR g=NOT PI TO x
  90 LPRINT CHR$ PEEK (f+g);: NE
XT g: LPRINT : NEXT f
```

Cassette Labeller

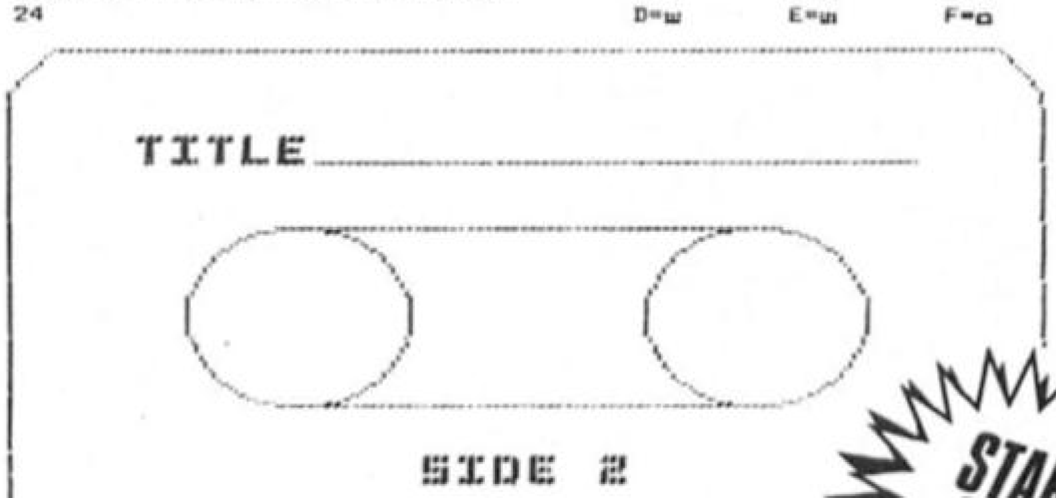
Finally for this month is a program from Lawrence Millea of Liverpool and it produces cassette labels for the actual cassette on a Sinclair or Alphacom or compatible printer.

The main problem he found in writing this program was that a correct sized label wouldn't fit on the paper, so he solved it by printing it sideways!

And so who gets this month's coveted Starcut award? (and twice the normal feel). Well this month we were unamimous in awarding it to Lawrence for his cunning routine, and tidying up of our office tapes!

Bye for now, and keep 'em coming!

```
   1 REM cassette labels
   5 GO SUB 1000
  10 PRINT AT 7,7; INVERSE 1;"CA
SSETTE LABELLER"
  20 INPUT  FLASH 1;"ENTER>"; FL
ASH 0;" THE CASSETTE SIDE NUMBER
 YOU REQUIRE (1 OR 2): ";A: IF A
<1 OR A>2 THEN  GO TO 20
  30 LET A$=CHR$ (149+A)
 100 CLS : RESTORE 190: FOR I=1
TO 2
 110 READ A,B,C,D,F,J,K,L
 120 PLOT 0,A: DRAW 0,B: DRAW 12
,C: DRAW 116,0: DRAW 0,D: PLOT 4
9,A: DRAW 0,F: PLOT 98,A: DRAW 0
,F: PLOT 31,J: DRAW 0,K: CIRCLE
73,L,25
 125 IF I=1 THEN  PRINT AT 21,14
;"ш";AT 19,14;A$
 130 IF I=2 THEN  PRINT AT 10,3;
"├";AT 9,3;"┤";AT 8,3;"┌";AT 7,3
;"┘";AT 6,3;"ш";AT 2,14;"ш";AT 1
,14;"┤";AT 0,14;"□"
 140 COPY : CLS : NEXT I
 190 DATA 0,106,10,-116,50,0,87,
51,175,-106,-10,116,-50,128,47,1
24
 200 INPUT "Do you want to print
 more labelsY or N ";b$: IF b$<>
 "y" THEN  STOP
 210 GO TO 10
1000 RESTORE 1010: FOR i=USR "a"
 TO USR "h"+7: READ x: POKE i,x:
 NEXT i: RETURN
1010 DATA 0,0,192,192,255,255,19
2,192,0,0,195,195,255,255,195,19
5,0,0,3,3,3,3,255,255,0,0,195,21
9,219,219,255,255,0,0,223,223,21
9,219,251,251,0,0,126,255,195,19
5,255,255,0,0,3,3,255,255,99,3,0
,0,243,251,219,219,219,223,79
1020>REM graphics chars.
        A=├        B=┤        C=┘
        D=ш        E=ш        F=□
```



STAR CUT

SHORTCUTS

RANDOM MEMORY

In the last column I sinned! I showed you how to USE UP memory instead of saving it, so for my penance I'll start this month with another byte-saving idea.

You may remember that in Part 1 I suggested saving space by holding the value of variables only in the variable area of the computer's memory (called VARS), and not in the program area (PROG) as well. For those who missed Part 1 the program line:

10 LET a = 5

takes up some 16 bytes of memory space when really all you want is the machine to know that the variable *a* has a value of 5. So type in, with no line number,

LET a = 5

and press ENTER. You'll get the O.K. message. Now LIST. Nothing! Not really surprising since you have no program, but the machine does know the value of *a*. You can prove this by typing:

PRINT a (ENTER)

and you'll get 5 printed on screen. I said then that you could only use this method for variables which didn't get changed, not for those which, for example, control strength, score, objects held etc in a games programe. Actually you can use the same idea for these provided that you reset them before you start again, having just lost your last life by bumping into a Balrog. What you do is write a short program to set up these variables. Something like

10 LET s=10 : LET l=5 : DIM f$(10,10)

where *s*, *l* and *f$()* are strength, lives and finds array. Now, RUN your program then delete all the lines and SAVE. Yes I know that sounds crazy, but what you'll be saving is just those initial value in VARS.

All you have to do now is MERGE this information with the main programe when you want to restart, with a line such as:

9999 PRINT "You're Dead! Rewind tape and play to try again" : MERGE "data" GOTO 10

Three points. First, SAVE the data programe on the adventure tape at the beginning. It makes it much easier to find when you want to reset. Secondly when you save the main program, make sure the data in VARS is already on board, and finally, don't ever use RUN. You'll lose all the data!

Graphics

Last month I left you with the problem of how to make larger illustrations with user defined graphics. We'll start this month with some solutions.

It is quite easy to enlarge the 5 by 4 character illustrations we've already made to double that size, provided you're prepared to accept a slight loss in resolution. (The sort of quality found in the "Dangermouse" adventures.) To do this you need to make use of the large characters routine found on your present from Uncle — your free Horizons tape. You need a copy of this to work on, loaded high up in memory so,

CLEAR 65090 : LOAD "c" CODE 65091 (ENTER)

and play side B of the tape. You'll get some programs and codes listed first, but after about two minutes the code you need will load in and you'll get the O.K. message. Don't be tempted to try it straight away or it will crash. The reason is that it uses the machine code instructions CALL and JUMP. Each of these is followed by an address to move to, but because the code is now much higher up in memory than it should be these jump instructions are not in the right places, and have to be corrected. Program 1 will do this for you. Type it in and RUN. It will find those incorrect addresses and correct them. It's worth saving a copy of this amended code with **SAVE "newcode" CODE 65091,277.** This is part of the code you must have on board when you print your illustrations.

Now for the udg data. You can design this as a 5 by 4 block as before, then use the 'Horizons' Character Generator program to convert each into code and SAVE it as explained last time. If you want something to work on, Table A gives the data for an illustration of a boot (and why not? Don't you always find them laying around in adventures?). Load this in, reading across each line, using

FOR f = 60000 TO 60159 : INPUT i : POKE f,i : PRINT f,PEEK f : NEXT f

You'll be using a statement similar to this a number of times so when, in the future, I refer to "codeloader" this is the statement I mean.

Save it with

SAVE "bootg" CODE 60000,160

Now, to try out the idea, type in Program 2. Let's have a look at what it's going to do before you RUN.

Line 10 POKEs into the printer buffer — a good place to store things — values that the "newcode" needs to use. The first two are the required magnification, horizontal and vertical, so the 2 in this case means twice normal size. The numbers 144 to 148 are the codes for the udgs A to E. The 255 is an end marker.

Line 20 sets up the value of *s*, the data start less 1152. The reason for the subtraction is worth an explanation. The routine looks in the ROM — the computer's own memory — for the shape of characters to print. You have to direct it to the udgs instead, but as the first udg is code 144 and there are eight bytes per character the routine has to be directed 144x8 (=1152) bytes below the real start. (If you're confused by that take



Clyde Bish returns with some more byte saving advice

consolation in the fact that I've only recently sussed it out!)

Line 30 does all the hard work. It resets the character pointer as explained above, sets the row and column pointer each loop, then calls the routine to print that row of udgs. Each loop prints successively lower rows, and Line 40 resets the character's pointer.

Now RUN and you'll see a big boot appear centre left of the screen. If you want the illustration elsewhere you must change the POKE 23306 value in line 30 to the number pixels across, and the first number in the POKE 23307 expression to the number of pixels up. The chart on page 76 of your manual will help.

In colour

If you want the picture in colour you'll need to add that information to the boot graphics data. Again you could use the utility I supplied last month, but I'll give you some to get on with. These are found in Table B and will produce a blue and red boot on yellow paper. Add these extra 20 bytes, one for each character square, with "codeloader", altering the FOR/TO values to 60160 and 61079.

You could get these onto the screen using a BASIC program as we have done before, but it would be rather slow. We would do better to resort to some machine code. What we need to do is take each data value, and POKE it into the attributes file in blocks of 2 by 2 character squares. Enter the data from Table C using "codeloader" with the FOR/TO values at 65046 and 65090. As this is added just before the "newcode" which is already present, you can save it all together with:

SAVE "gcode" CODE 65046,322

Note that the routine assumes that the position of the top left character is row 10, column 0. You can change this with:

RANDOMIZE row *8 + column + 22528 : POKE 65050,PEEK 23670 : POKE 65051,PEEK 23671

To use this extra code make the following changes to program 2:

Line 30 — add INK 7 to the start of the line, and append INK 0 to the end.

Line 40 — add to the end of the line, **RANDOMIZE s + 1312: POKE 65047, PEEK 23670 : POKE 65048, PEEK 23671 : RANDOMIZE USR 65046**

Now RUN and you'll find a multi-coloured boot suddenly appears on screen. The reason for the sudden appearance is the INK 7 . . . INK 0. The boot is drawn in white INK on white PAPER and so is invisible until the colour is added.

You could of course, use these ideas to produce other illustrations. A double size 7 by 3 illustration top centre could provide a location scene with text underneath. If you understand the system so far you'll find it quite easy to make the changes. (If not, drop me a line at Golden Square and I'll give you some help.)

Now what about those perfectionists who want large pictures plus maximum resolution? No problem — except the amount of memory they'll use up. A full screen colour illustration uses up 6336 bytes, so you could only get five in the top 32K of memory. This may be O.K. if the controlling program is quite simple, and I'll show you how to design your own "Mugsy"-type game in the next issue, but for the moment let's stick with smaller fry. How about a 16 by 8 character illustration top centre, with space either side for scores, strength etc., and scrolling text underneath?

Pictures

First the pictures. You've obviously get to design these yourself. (Try two simple ones first. Leave your upgrade of "Runestone" until later!) Once you have these drawn out on a suitable pixel/character square grid help is at hand to store and colour them. Load in the "character generator" program and convert each character square along the top row to an udg, then save these to tape (remember you only want A to P). Repeat the process for the second and subsequent rows and repeat for the second picture, saving each of the codes in order, one after the other on the tape. Now you need a utility to store them as complete pictures in high memory, and also add colour information. Program 3 is what you need. Type it in, SAVE for future use then RUN. Answer the first prompt with the number of pictures to store (in this case two). The program clears the required space and tells you the address of the start of the first picture. Make a note of this (and subsequent addresses), as you'll need them later. You'll then be invited to play the data tape you made with those blocks of 16 udgs. The program will load these in to the correct addresses. When the eighth block of a set is loaded, stop the tape and add the colour. You'll see the picture displayed with the first character square flashing. Input the attribute for this and subsequent squares. You can work them out using the table in the last issue (or by INK + PAPER * 8 + (128 for FLASH) + (64 for BRIGHT) if you don't have it — That'll teach you!)

If the attribute is the same as the previous square just press ENTER. When all 128 have been filled in you'll move to the next picture to repeat the process, then save the combined code to tape, ready to load back in at the correct address.

To use this data you'll need a short machine code routine to drop it down instantly to the Display and Attribute Files. Use "codeloader" to enter the data from Table D, changing the FOR/TO values to 65324 and 65367, then save as

SAVE "downcode" CODE 65324,44.

I'd like to explain how the routine works but space is fast running out so I'll press on and give you the routine you'll need to have in your program to make use of it. One important point though, you must set the variable **n** to the start of the required picture data (noted from Program 3) before calling it. Assuming that you did just save two pictures, Program 4 will allow you to switch instantly between them. Type this in and try it out (making sure you still have your picture data and "downcode" on board). Fast, wasn't it?

Table A

0	0	0	0	15	8	8	15
0	0	0	7	255	2	2	255
0	0	224	224	224	96	96	240
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
8	15	15	15	15	15	31	31
1	255	255	255	255	255	255	255
48	48	240	48	48	248	152	152
0	0	0	0	0	0	7	59
0	0	0	0	0	224	240	248
31	31	63	63	63	63	63	63
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
251	251	253	254	255	255	255	255
252	252	254	254	2	254	254	254
63	32	63	16	15	0	0	0
255	1	255	1	255	0	0	0
3	248	7	0	0	0	0	0
248	7	240	15	0	0	0	0
2	252	4	248	0	0	0	0

Table B

50	50	49	49	49	49	49	49
49	49	49	49	49	49	49	50
50	49	49	49				

Table C

33	0	235	17	64	89	14	4
6	5	197	126	18	19	18	6
31	19	16	253	18	19	237	160
6	32	27	16	253	193	16	234
197	6	54	19	16	253	193	13
120	177	32	220	201			

Table D

33	0	125	17	8	64	62	0
1	16	0	237	176	6	16	19
16	253	60	254	64	56	241	17
8	88	62	0	1	16	0	237
176	6	16	19	16	253	60	254
3	56	241	201				

Now for the promised scroll. This is called from the ROM using RANDOMIZE USR 3583. It scrolls only the bottom 11 lines, so anything on the top half of the screen (your illustration for example) is quite safe.

Add the following lines to Program 4:

15 RANDOMIZE USR 3583 : PRINT AT 21, 0; "This is picture 1"
25 RANDOMIZE USR 3583 : PRINT AT 21, 0; "This is picture 2"

then change between them a few times and you'll see the effect.

That's all for this issue. Tune in next month for the Big Time, including a way of holding a whole screen colour illustration in less than 200 bytes!

PROGRAM 1

```
   1 REM "c"CODE from "HORIZONS"
     tape must be loaded to 65091
     after CLEAR 65090
  10 LET s=65091
  20 DEF FN a(x)=INT ((s+x)/256)
  30 DEF FN b(x)=s+x-256*FN a(x)
  40 RESTORE : FOR i=1 TO 5: REA
D y,x: POKE s+y,FN b(x): POKE s+
y+1,FN a(x): NEXT i
  50 DATA 106,32,127,164,86,3,15
4,48,251,156
```

PROGRAM 2

```
   1 REM You must have on board
   "newcode" and "bootg" codes
  10 POKE 23308,2: POKE 23309,2:
 POKE 23310,8: POKE 23311,144: P
OKE 23312,145: POKE 23313,146: P
OKE 23314,147: POKE 23315,148: P
OKE 23316,255
  20 LET s=58848: REM start of g
raphics data less 1152
  30 FOR f=0 TO 3: RANDOMIZE s+4
0*f: POKE 23606,PEEK 23670: POKE
 23607,PEEK 23671: POKE 23307,81
+f*18: POKE 23306,0: RANDOMIZE U
SR 65091: NEXT f
  40 POKE 23606,0: POKE 23607,60
```

PROGRAM 3

```
  10 INPUT "No. of pictures to s
tore ";p: POKE 23681,p
  20 LET d=65367-(p*1152+44): RA
NDOMIZE d: POKE 23728,PEEK 23670
: POKE 23729,PEEK 23671: CLEAR d
-1: LET d=256*PEEK 23671+PEEK 23
670: LET s=d: LET p=PEEK 23681:
FOR i=1 TO p: CLS
  25 PRINT "Picture ";i;" starts
 at ";d
  30 PRINT '"PLAY TAPE"
  40 FOR f=1 TO 8: PRINT AT 4,0;
"Loading row ";f: LOAD ""CODE d+
(f-1)*128
  60 NEXT f: PRINT '"STOP THE TA
PE": PAUSE 200: CLS
  70 LET c=0: FOR f=d TO d+1020
STEP 128: RANDOMIZE f: POKE 2367
5,PEEK 23670: POKE 23676,PEEK 23
671: PRINT AT c,8;"ABCDEFGHIJKLM
NOP": LET c=c+1: NEXT f:
REM Caps in quotes are udgs.
Enter in G mode
  80 LET d=d+1024: FOR f=22536 T
O 22760 STEP 32: FOR n=f TO f+15
: POKE n,135
  90 INPUT "attribute? ";a$: IF
a$="" THEN POKE n,a: GO TO 110
 100 LET a=VAL a$: POKE n,a
 110 POKE d,a: LET d=d+1: NEXT n
: NEXT f
 120 NEXT i
 130 INPUT "Title for code? ";a$
 140 SAVE a$CODE s,p*1152
```

PROGRAM 4

```
   1 REM You must have "downcode
"      on board
  10 LET n=63019: GO SUB 9999: I
NPUT "Press enter";a$
  20 LET n=64171: GO SUB 9999: I
NPUT "Press enter";a$
  30 GO TO 10
9999 RANDOMIZE n: POKE 65325,PEE
K 23670: POKE 65326,PEEK 23671:
RANDOMIZE USR 65324: RETURN
```

AVOID THE MONSTER RUSH FOR



SUBSCRIPTION RATES

£15.00 for 12 issues U.K.
£18.00 for 12 issues Overseas Surface Mail
£61.20 for 12 issues Overseas Airmail

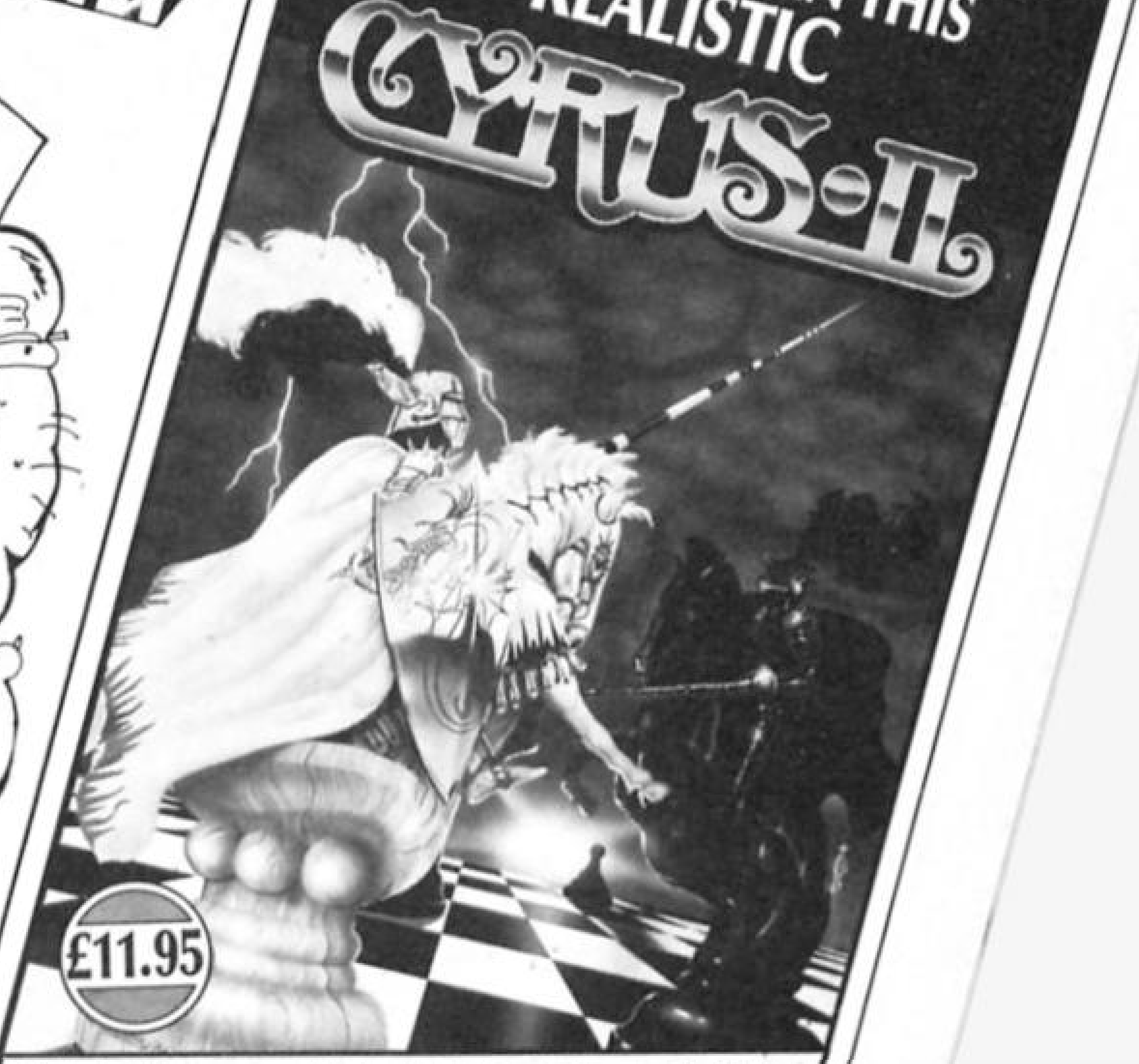
Please begin my subscription(s) to ZX COMPUTING MONTHLY with the .... issue.
I enclose my cheque/postal order for £ .... made payable to Argus Specialist Publications Ltd. or debit £ .... from my Access/Barclaycard No. [ ] valid from .... to ....

NAME (Mr/Mrs/Miss) ....
ADDRESS ....
....
Postcode ....
....
Signature ....
Date ....

Please use BLOCK CAPITALS and include post codes

Send this form with your remittance to:
INFONET LTD., Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB.

CHESS ON THE SPECTRUM HAS NEVER BEEN THIS REALISTIC



Playing is believing as far as this the ultimate chess playing sensation for the Sinclair Spectrum is concerned. The many features of this superior version means it will beat all existing Sinclair Spectrum chess programs published.

- *Variable level performance*
- *Set pieces*
- *1900 standard book opening*
- *Problem solving*

plus many, many more unique features....

From beginner through to tournament playing expert Cyrus II is the only chess program that should interest you.

YOU DON'T BELIEVE US?!

Send us your old Spectrum chess program and we'll send you Cyrus II for the special offer price of **£6.95.** If you are not completely satisfied we'll refund your money along with your old chess game. **Mark 1 – Spectrum 48K Mark II – Spectrum 128K**

Please send me .... copy/copies of Cyrus MKI [ ]* MKII [ ]*
I wish to take advantage of your special offer and enclose my old Spectrum chess program (delete if not applicable).

Please find enclosed a cheque/P.O.** for the sum of £ ....

Charge my Access/Visa the sum of £ ....

Card No. .... Signature ....

Name .... Address ....

.... Postcode ....

*Tick as required ** Made payable to Alligata Software Ltd.
Please allow 7 days for delivery.
Send to: **Alligata Software Ltd., 1 Orange Street, Sheffield, S1 4DW.**

ALSO AVAILABLE... MARK II 3D CYRUS II FOR SPECTRUM 128K





Listing 1

```
100 CLS#0
110 hs=0
120 REPeat start
130 f=20
140 level=0
150 s=0
160 REPeat ny_tur
170 PAPER 0:CLS
180 x=10:y=35
190 old$=CHR$(244)
200 a=1
210 b=1
220 v=1
230 w=2
240 q=4
250 u=0
260 t=14
270 tid=0
280 rita
290 REPeat program
300 gubbe
310 koll_u
320 koll
330 IF u<2 THEN spoke
340 IF x=a AND y=b AND u=1 THEN ta
350 time
360 END REPeat program
370 REMark #### rita ####
380 DEFine PROCedure rita
390 REMark #### rita ####
400 IF f<120 THEN f=f+5
410 DIM o(f),p(f),e(19,36)
420 FOR j=1 TO f
430 BEEP 100,j
440 o(j)=RND(1 TO 18)
450 p(j)=RND(1 TO 34)
460 krock=e(o(j),p(j))
470 IF krock=1 THEN GO TO 440
480 IF o(j)<10 AND p(j)<2 THEN G
O TO 440
490 e(o(j),p(j))=1
500 PAPER RND (1 TO 7)
510 AT o(j),p(j):PRINT " "
520 END FOR j
530 PAPER 3
540 FOR j=0 TO 19
550 AT j,0:PRINT " "
560 BEEP 10000,j+5
570 AT j,36:PRINT " "
580 END FOR j
590 FOR j=36 TO 0 STEP -1
600 AT 0,j:PRINT " "
610 IF j<=19 THEN BEEP 10000,j+5
620 AT 19,j:PRINT " "
630 END FOR j
640 INK 7
650 AT 19,2:PRINT "TIME "
660 FOR j=0 TO 7
670 PAPER j
680 AT 19,j+7:PRINT " "
690 END FOR j
700 PAPER 3
710 AT 19,16:PRINT "SCORE ";s
720 AT 19,27:PRINT "HS ";hs
730 AT 0,18:PRINT "BLOCKS ";f
740 level=level+1
750 AT 0,7:PRINT "LEVEL ";level
760 m=RND(1 TO 18)
770 n=RND(1 TO 35)
780 krock=e(m,n)
790 IF krock=1 THEN GO TO 760
800 PAPER 0
810 AT m,n:PRINT "*"
820 rita_bur
830 END DEFine rita
840 REMark #### gubbe ####
850 DEFine PROCedure gubbe
860 REMark #### gubbe ####
870 PAPER 0
880 AT x,y:PRINT " "
890 ny$=INKEY$
900 IF ny$="" THEN ny$=old$
910 IF ny$=CHR$(192) THEN y=y-1
920 IF ny$=CHR$(200) THEN y=y+1
930 IF ny$=CHR$(208) THEN x=x-1
940 IF ny$=CHR$(216) THEN x=x+1
950 old$=ny$
960 INK 7
970 AT x,y:PRINT "?"
980 GO TO 310
990 REMark #### koll ####
1000 DEFine PROCedure koll
1010 REMark #### koll ####
1020 krock=e(x,y)
1030 IF krock=1 THEN tut:slut
1040 IF x=0 OR x=19 THEN tut:slu
t
1050 IF y=0 OR y=36 THEN tut:slu
t
1060 END DEFine koll
1070 REMark #### time ####
1080 DEFine PROCedure time
1090 REMark #### time ####
1100 tid=tid+1
1110 IF tid=15 THEN
1120 PAPER 3:AT 19,t:PRINT " ":t
id=0:t=t-1
1130 END IF
1140 IF t=6 THEN tut:slut
1150 IF t=7 THEN BEEP 100,1
1160 END DEFine time
1170 REMark #### spoke ####
1180 DEFine PROCedure spoke
1190 REMark #### spoke ####
1200 wv
1210 IF w=0 OR w=19 THEN GO TO 1
340
```

CATCHER

by Tomas Jonasson

D.I.Y.

Capture the ghouls in this ghostly maze game for the QL.

Avoid the blocks in the maze or you'll find yourself joining the ghosts in limbo. Your aim is to get the key, capture the ghost and despatch him to prison.





```
1220 IF v=0  OR v=36 THEN GO TO
1340
1230 krock=e(w,v)
1240 IF krock=1 THEN GO TO 1340
1250 PAPER 0
1260 AT a,b:PRINT " "
1270 IF q=1 THEN b=b-1
1280 IF q=2 THEN a=a-1
1290 IF q=3 THEN b=b+1
1300 IF q=4 THEN a=a+1
1310 PAPER 7:INK 0
1320 AT a,b:PRINT CHR$(34)
1330 END DEFine spoke
1340 REMark **** fel ****
1350 q=RND(1 TO 4)
1360 END DEFine
1370 REMark **** wv ****
1380 DEFine PROCedure wv
1390 REMark **** wv ****
1400 w=a:v=b
1410 IF q=1 THEN v=v-1
1420 IF q=2 THEN w=w-1
1430 IF q=3 THEN v=v+1
1440 IF q=4 THEN w=w+1
1450 END DEFine wv
1460 REMark **** slut ****
1470 DEFine  PROCedure slut
1480 REMark **** slut ****
1490 INK 4:CSIZE 3,1:FLASH 1:PAP
ER 6:CLS
1500 AT 0,7:PRINT "GHOST CATCHER
"
1510 PAPER 0:INK 7:FLASH 0
1520 AT 0,5:PRINT CHR$(34)
1530 AT 0,21:PRINT CHR$(34)
1540 INK 7:PAPER 6
1550 AT 2,2:PRINT "SCORE ";s
1560 IF s>hs THEN LET hs=s:FLASH
 1
1570 AT 4,2:PRINT "HI SCORE ";hs
1580 FLASH 0
1590 AT 6,2:PRINT "LEVEL ";level
1600 AT 8,2:PRINT "PRESS ANY KEY
 TO PLAY"
1610 INK 0:CSIZE 2,0
1620 AT 19,7:PRINT "**** Tomas J
onasson ****"
1630 FOR j=1 TO 400:END FOR j
1640 REPeat till_start
1650 IF INKEY$="" THEN ELSE END
REPeat start
1660 END REPeat till_start
1670 REMark **** koll_u ****
1680 DEFine PROCedure koll_u
1690 REMark **** koll_u ****
1700 SELect ON u
1710 ON u=0
1720 IF x=m AND y=n THEN
1730 PAPER 3:INK 7
1740 FOR j=1 TO 10
1750 BEEP 10000,j
1760 s=s+1
1770 AT 19,22:PRINT s
1780 END FOR j
1790 u=1
1800 END IF
1810 ON u=1
1820 IF x=a AND y=b THEN ta
1830 ON u=2
1840 IF x=1 AND y=35 THEN
1850 PAPER 3:INK 7
1860 FOR j=1 TO t
1870 s=s+1
1880 AT 19,22:PRINT s
1890 BEEP 10000,j
1900 END FOR j
1910 END REPeat ny_tur
1920 END IF
1930 END SELect
1940 END DEFine
1950 REMark **** tut ****
1960 DEFine PROCedure tut
1970 REMark **** tut ****
1980 FOR j=40 TO 1 STEP -1
1990 BEEP 10000,j+10
2000 BEEP 10000,j
2010 BEEP 10000,j+20
2020 END FOR j
2030 BEEP 1,1
2040 END DEFine tut
2050 REMark **** rita_bur ****
2060 DEFine PROCedure rita_bur
2070 REMark **** rita_bur ****
2080 RESTORE 2150
2090 FOR j=1 TO 8
2100 READ ba,bb,bc,bd
2110 LINE ba,bb TO bc,bd
2120 END FOR j
2130 END DEFine rita_bur
2140 REMark **** data ****
2150 DATA 156,91,156,100
2160 DATA 156,100,165,100
2170 DATA 165,100,165,91
2180 DATA 165,91,156,91
2190 DATA 159,91,159,100
2200 DATA 156,94,165,94
2210 DATA 156,97,165,97
2220 DATA 162,91,162,100
2230 REMark **** ta ****
2240 DEFine PROCedure ta
2250 REMark **** ta ****
2260 PAPER 3:INK 7
2270 FOR i=10 TO 1 STEP -1
2280 BEEP 10000,i
2290 s=s+1
2300 AT 19,22:PRINT s
2310 END FOR i
2320 u=2
2330 END IF
2340 END DEFine ta
```

GAMES

Captain Slog

Alpha-Omega Software
£2.99

This game is of the maze, chase, search and find variety and a most acceptable one at that. The cassette insert instructions are beautiful in their brevity. I quote:

"There are three types of barrier key, red green and white. Each key allows you to pass through the corresponding colour barrier. The nine blue-prints are spread out into different sectors within the game. You have to collect all nine before you are allowed access to the teleport room."

There are several other objects which you may collect to increase your score, others will give you more energy or use it up. Avoid all contact with robots".

Given just this I would not be inspired to open the case let alone load the program, and that would have been a shame because for me it had that perfect blend of difficulty and challenge that makes a game addictive. OK, so it is an old format, similar to many others, the animation is pretty simple, enemy sprites are few in variety and sound is fairly elementary. But for me it works.

Much as I love pitting wits with a programmer who has taken great pains to devise a difficult, devious game, I also love relaxing with a straightforward five key exercise in pure reactions.

The five keys in this game control L/R/U/D and fire. Collecting an object is done by getting next to it, screens are entered and exited by wandering off the edge and a new one is presented, no scrolling or tiresome fades.

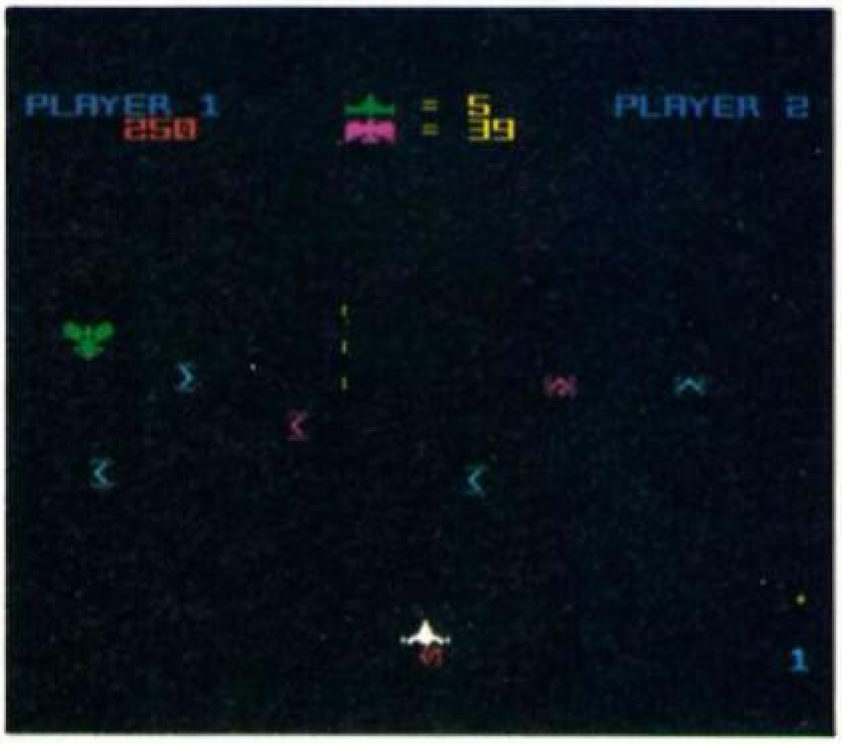
Mapping is a possibility as blue-prints and keys appear to be in the same locations, certainly the maze is not randomly formed, personally this is too much like hard work for a program which I intend to use for simple pleasure, but anyone who desires to take it more seriously would be able to do so.

Locations vary and are nicely designed, watch out for the blob which simply blocks your way, it can trap you and then it's a case of end of game. You are only given one life and not much energy.

Worth adding to your collection as long as you do not expect a state of the art program.







Star Firebirds

Firebird
£1.99

I well remember reviewing this game when it was sent to us by Insight and saying that it was a good game but overpriced. Either they harkened or sold out to Firebird.

It remains a good game and more realistically priced at £1.99, so much so that I have no hesitation in recommending it to Zap-Em-All fans everywhere.

You may have gathered from the above that this is a game in the "if it moves, shoot it" class, and so it is. Forget the insert blurb, the game consists of wave after wave, each comprising of 50 "birds", of aliens attacking you and you have to zap them quickly as they circle, swoop and dodge. As you progress through each wave the action gets faster and other problems occur, such as scatter bombs!

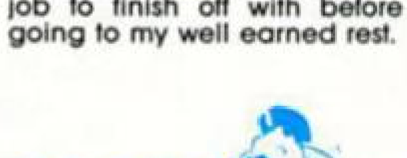
The animation is simple but effective, the birds appear and get larger as they follow their patterns and you move left and right and blast away at them as fast as you can.

Just to make life that extra bit difficult they will sometimes appear from beneath you or fly along the bottom of the screen making the task of avoiding them awkward, if not impossible.

Of course they are not helpless sitting pigeons, they drop bombs at you and contact with one is fatal, your five lives are soon depleted. Sound is limited to a few burps during actual play and a respectable intro tune, colour is bright, clear and doesn't clash, and movement is smooth.

A range of options including joysticks and starting levels is provided and a full high score "hall of fame" is included. It is very easy to beat the scores which start the Hall of Fame, but no problem, your own sets can be saved and reloaded if you wish.

A nice unpretentious mindless killing game, and I'm not knocking it, after a day of reviewing the latest mind stretching arcventure it's just the job to finish off with before going to my well earned rest.



Full Throttle

Elite Classics
£2.99

It had passed through my mind at some time that there was a lot of life left in some of the earlier games for the Spectrum and now Elite has seen fit to market what to us old hands are indeed classics.

This program was originally produced by Micromega and received much acclaim in its time, so how has time treated it?

Before going into personal opinion I think a description of the game is in order. It is a motor cycle racing simulation in perspective 3D and you have control over left/right turn and lean, accelerate and brakes either via the keyboard or joystick. You can choose to race any one of ten circuits, all of which I am assured are authentic representations, selecting the number of laps and practice or race mode.

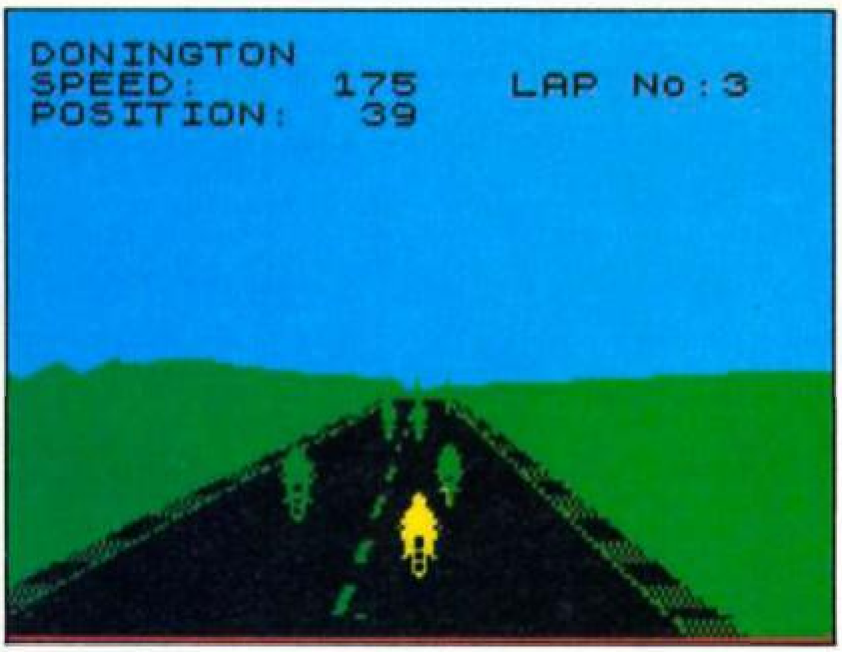
Your view is of the track stretching out from the bottom of the screen and providing you are racing, other bikes disappear or loom larger as you are overtaken or overtake other riders.

In the sky, area information is given on the track, lap number, speed and position. You will skid if you take a corner too fast, slow down if you run off the track and stop if you collide with any of the other rides, a frequent event for me as they weave around unpredictably.

There is a bit of flicker, especially on the other rides and some colour clash when your cycle goes off the track, sound is a rather high pitched blipping which, with a bit of imagination passes for an engine noise. None of this spoils the game to any real extent and you soon get absorbed in trying to beat those computer controlled machines, all 39 of them!

It was great to replay this game, one of my old favourites, the original of which lurks in the depth of my program box, and it made me wish I'd kept it closer to hand. No it hasn't lost any of its magic and I'd still recommend it to you even if it was still at the original price. At £2.99 it's a snip if you haven't already got it.







Cagara

Players
£1.99

You have been teleported to the land of Cagara by managing to get yourself caught in an evil gypsy's crystal ball. Presumably, you crossed her palm with cupro-nickel instead of silver. The only way you can regain your freedom is to explore the castles, mazes and jungles in Cagara and find the magic chalice. A reasonable if hardly original storyline. An element of originality is brought into play by the fact that you are accompanied on this adventure by your sulky pet hamster called Eric who for some strange reason has sprouted helicopter blades!

Not that you would notice, for he is depicted on screen as a small red blob with spikes on who always seems to run off the first screen, never to be seen again. This is a trifle unfortunate as he is supposed to fight your battles for you. Instead, when you (represented on screen by a medium sized blob) are chasing through a maze and a monster appears (a big blob) you tend to get killed automatically as it can fly through hedges and you can't. One contact by a monster and it is curtains for you — game over time. There are other ways of defending yourself. Scattered around the grounds are a number of objects. You can only carry one of these at a time and must work out which thing does what. This is made more difficult for you as the items are such small blobs that they are instantly unrecognisable. The only one that looked like anything was a pink castle that teleported you to another castle.

The rooms have equally non stunning graphics, the highlight of the jungle chambers being whether they have two, three or four exits. The game looks as if it was written when the Spectrum first came out. Cagara is so bad, it is almost worth buying for that very reason.



I.C.U.P.S.

Thor
£8.95

In case you are wondering I.C.U.P.S. stands for the International Commission for Universal Problem Solving. Now you know that you can forget it immediately. The inlay blurb blathers about your special mission etc but doesn't even claim that the game is real. You are not 'at the controls of a Hypermegathrust-strike fighter' as per usual in this type of game, instead it rather tamely tells you that these are computer games to test your skill and decide whether you can become an agent for I.C.U.P.S.

In short what we have here are two unrelated bits of game stuck together with a poor attempt at giving it some sort of unifying idea. First off is a brisk shoot em up in three sections of space tunnel. The mothership dumps you and you have to blast/avoid/outrun the enemy ships and get to the end where you enter phase two controlling a "Hostile Environment Anthrobot" who must find four parts of bomb hidden in his starship. The robot glides nicely through a rather unpopulated maze of decks but even the inevitable nasties are halfhearted and don't seem too intent on inflicting damage.

This is a very poor product for a full price game which seems to have been thrown together with the thought that if one tired formula idea won't work maybe two will. Unfortunately it doesn't and emphasising that you are just playing another computer game is surely a mistake. Aren't computer games about creating illusions? Who wants to be play a game about playing a computer game? You guessed it.





Virgin Atlantic Challenge Game

Virgin
£5.95

You've to to admire the man, if there's an angle Richard Branson will exploit it, I suppose you don't get to be a millionaire without that kind of acumen anyway.

It's been some time since I saw a Virgin computer game but, after a shaky start, their products have been quite respectable, so, is this a bit of Branson genius or a right pickle?

Well the game itself is of the Strategy type with the nice frill of a graphic action sequence to lift it out of the usual run of the mill rut. You take the role of Richard (a role I've been trying to emulate in real life for ages!) on his much publicised and triumphant attempt to cross the Atlantic in record time.

You control a recognisable graphic of him along the deck of the Challenger to the Telex, Radar, Galley, Map or Engine room displays. He can also be made to climb into the cockpit to alter direction.

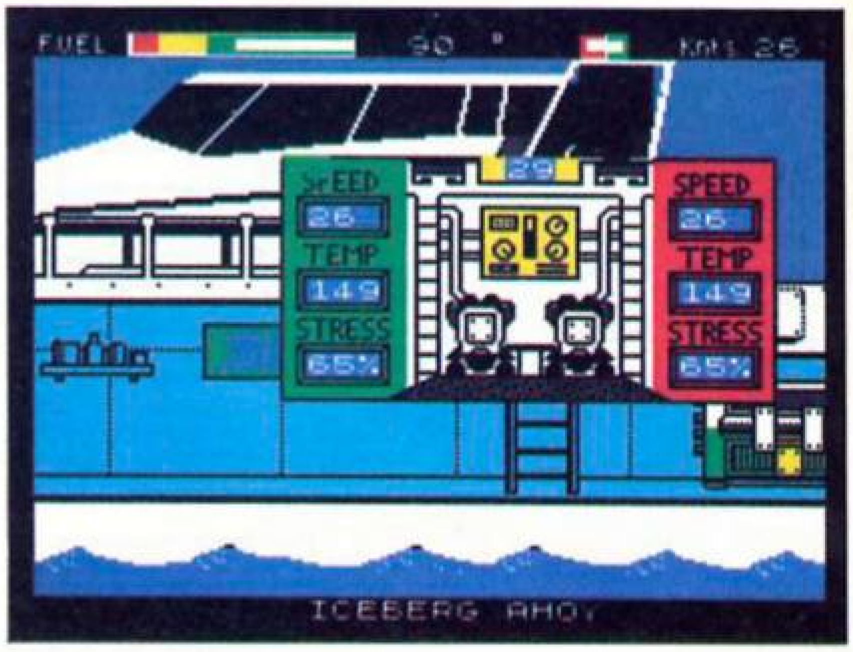
As you may gather, this is not a leisurely decision making exercise but a mad dash around the screen in response to computer messages to adjust the various parameters of direction and speed, eat and to make lightning business decisions to telexed questions.

I was sure he had other crewmen aboard to help him with all of this but in this game he does it all on his own, he really ought to learn how to delegate! If this is a realistic simulation of his day he'll have a heart attack soon.

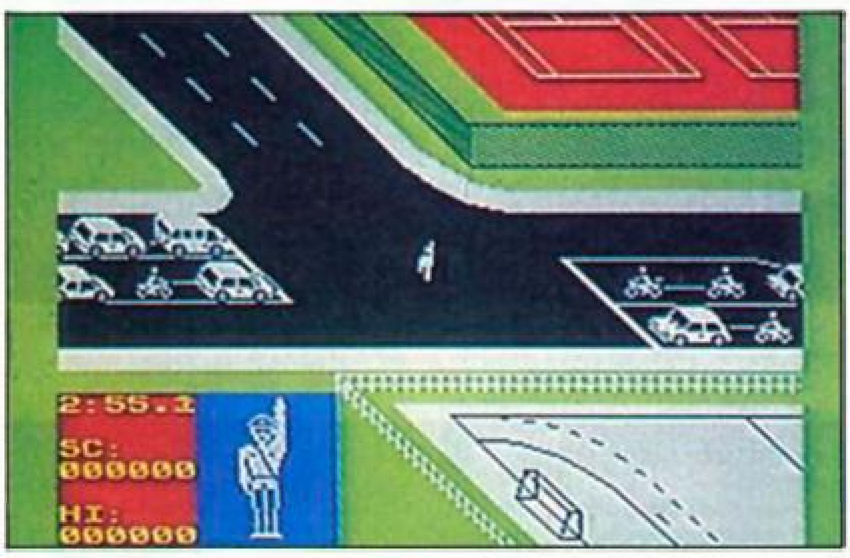
No it's not easy and I found myself getting quite irritated with it, the number of fishing boats and icebergs floating around seems unrealistically high, also in joystick mode the responses seem rather slow at times.

Probably my biggest moan is the lack of high score table, I know I was getting better scores:— 5% and 66 at the time of writing, but it is nice to have previous ones on screen as a reminder.

Finally I must admit to enjoying playing it and if this kind of mental gymnastics is your cup of tea then you'll like it too.







Sbugetti Junction

Bugbyte
£2.99

Luigi is a trainee traffic cop and has come all the way from Macaroni to learn how we Brits cause accidents over here. Before he can return to his wife Fettucine, he has to let lots of cars go pasta him without keeping anyone waiting too long.

The screen is divided into two areas. The junction full of assorted bikes, cars and lorries wanting to go every which way occupies the main area whilst in the bottom left hand corner of the screen, there is a large picture of Luigi doing semaphore impressions and another box giving details of how well he is doing them. There are plenty of game options to select from ranging from degrees of difficulty (light, heavy and rush hour traffic) to whether cars drive on the left or right side of the road.

Luigi can turn to face any of the four directions and can then wiggle his arms around to give various signals such as stopping a lane of traffic or allowing another lane to turn right. Everything must be kept moving smoothly and no-one should have to wait too long before being allowed to move. Dither too much and the cars in the lane in question start to flash. Unless you now remedy things straight away, the game ends. Points are awarded for each car successfully through the junction. Get everyone through without crashing or waiting and you move to the next, more difficult junction.

Sbugetti Junction is a pleasant enough game to play but doesn't really have any lasting appeal. Unless you have always wanted to be a policeman or traffic warden or the Mafia have made you an offer that you cannot refuse, give this one a miss.



Ghostbusters

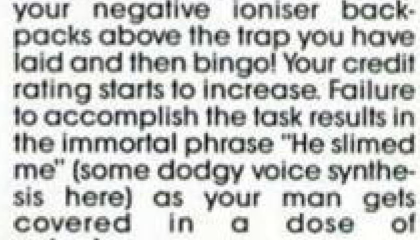
Activision
£9.99
Spectrum 128

Who you goin' to call? — GHOSTBUSTERS! Not exactly a haunting tune but rather one more likely to give you nightmares as you find that you can't get it out of your brain. Now, the game of the film has been released in an enhanced version for the Spectrum 128. The added features include better music and graphics but the game remains the same.

For anyone who has been away for the last couple of years and doesn't know the storyline, New York is suffering from an invasion of ghosties, ghoulies and things that go bump in the night. You are setting up a franchise to make as much money as you can by catching ghosts and stopping them reaching the Temple of Zuul where the climax of the game takes place. First, you must buy and equip your car. There are four basic models to choose from with speed and amount of equipment that can be carried increasing with price. The bits and pieces that you can buy include ghost vacuums and traps, image intensifiers and marshmallow sensors.

Having equipped your vehicle, you must then drive round the streets catching "roamers" and stopping in front of buildings where a slimer is present. By careful positioning of your men, you must manoeuvre the slimer using your negative ioniser backpacks above the trap you have laid and then bingo! Your credit rating starts to increase. Failure to accomplish the task results in the immortal phrase "He slimed me" (some dodgy voice synthesis here) as your man gets covered in a dose of ectoplasm.

Despite the vast amount of hype that preceded this game on its initial release and despite the fact that it is one of the best selling computer titles ever, Ghostbusters is not a particularly good game and is certainly beginning to show its age now. Even with the added extra bits on the 128, it is still not very good value for just under ten pounds.







SINCLAIR TITLES

SOFTWARE

POLE POSITION	£3.95
KNOCKOUT	£2.95
MATCH FISHING	£2.95
SON OF BLAGGER	£1.99
JET SET WILLY	£2.95
BC's QUEST FOR TIRES	£3.95
ASHKERON	£2.50
CODENAME MATT II	£1.99
EUREKA	£1.99
BLOCKBUSTERS	£3.95
TRAIN GAME	£2.95
GREMLINS	£2.95
HELLFIRE	£2.95
HIGHWAY ENCOUNTER	£3.95
ALIEN HIGHWAY	£3.95
T.L.L.	£2.95
MOON ALERT	£1.99
ROLANDS RAT RACE	£3.95
DALEYS DECATHLON	£2.95
DALEYS SUPERTEST	£4.95
SOLD A MILLON INC. DECATHLON, SABRE WULF, BEACHEAD, T.S.W.	£4.95
ARCADE HALL OF FAME INC. RAID II, BLUE MAX, ROCCO, FLAK, HUNCHBACK II	£4.95
FOOTBALL MANAGER	£3.50
MINI OFFICE	£3.50
IMPOSSIBLE MISSION	£4.95
TASWORD II	£4.95
BOUNTY BOB	£3.95
BEACHEAD II	£4.95
BRUCE LEE	£3.95
LINKWORD SPANISH	£2.50
NEVER ENDING STORY	£4.95
BEACHEAD	£2.95
ROBOT MESSIAH	£2.50
TECHNICIAN TED	£2.95
AVALON	£2.50
ACTIVE COMPUTER LEARING BOOKS + 4 TAPES	£2.95
MATCHPOINT	£3.95
NOW GAMES INC. LORDS OF MIDNIGHT, FALCON PATROL, ARABIAN NIGHTS, BRIAN BLOODAXE, STRANGELOOP, PYJAMARAMA	£3.99
ASTRO CLONE	£3.95
LAZY JONES	£2.50
WORTED MONTY MOLE	£2.95
MOON CRESTA	£3.50
DARK STAR	£2.95
LORD OF THE RINGS	£6.95
EVERYONES A WALLY	£2.95
JONAH BARRINGTONS SQUASH	£2.95
BACKPACKERS GUIDE	£1.99
WIZARDS LAIR	£2.50
WHEELIE	£3.50
SUBTERRANEAN STRYKER	£1.99
JASPER	£1.99
VU CALC	£2.50
CHESS BY SINCLAIR	£1.99
CHEQUERED FLAG	£1.99
EMERALD ISLE	£4.50
ADVENTURE QUEST	£4.50
LORDS OF TIME	£4.50
SNOWBALL	£4.50
SABRE WULF	£2.50
BIZZICOM (BUSINESS CONTROL SYSTEM)	£2.95
PAC MAN BY ATARISOFT	£5.95
VALHALLA	£2.50
TWIN KINGDOM VALLEY	£2.95
LINWORD GERMAN	£2.50
PSYCHEDELIA	£1.99
QUICKSILVA WORD PROC.	£1.99
BRAXX BLUFF	£1.99
DEATH CHASE	£1.99
CODENAME MATT	£1.99
HEATHROW A.T.C.	£2.95

EDUCATIONAL PACKS

PHYSICS 'O' LEVEL and 'A' LEVEL, MATHS 'O' LEVEL, CHEMISTRY 'O' LEVEL, BIOLOGY 'O' LEVEL	£9.95
COLLINS 4-11 YEARS PACK WHATS THE TIME, KNOW YOUR TABLES, FIRST NUMBERS	£7.95
COLLINS PADDINGTON PACK PADDINGTONS EARLY VISIT, PADDINGTONS GARDEN GAME, PADDINGTON AND THE DISAPPEARING INK	£7.95
HILL McGIBBON PACK PUNCTUATION PETE, BALLOONING, CAR JOURNEY, SPECIAL AGENT	£7.95
LONGMAN SOFTWARE 6-10 YEARS SUM SCRUNCHER, ROBOT RUNNER, WILD WORDS, A.B.C. LIFT OFF	£6.95
MR T PACK INC. MR T's SIMPLE SUMS, MR T MEETS HIS MATCH, MR T's ALPHABET GAME	£5.95
MR T PACK INC. MR T's NUMBER GAMES, MR T's SHAPE GAMES, MR T's MEASURING GAMES	£5.95
GRIFFIN SOFTWARE PACK INC. 5 YEARS AND OVER WORDSPELL, FAIRSHARE, WORDGAMES	£5.95
GRIFFIN SOFTWARE PACK INC. 5 YEARS AND OVER TABLESUMS, NUMBERFUN, GET SET	£5.95
GRIFFIN PYTHAGORAS PACK INC. 10-15 YEARS THEORUM OF PYTHAGORAS, FORMULAE & COMPOUNDS, FUNDAMENTAL ALGEBRA	£5.95
GRIFFIN MATCHS & CHEMISTRY PACK 10-15 YEARS ATOMS & MOLES, INTRO TO TRIGONOMETRY, MENTAL ARITHMETIC	£5.95

GAMES PACKS

SILVERSOFT PACK INC. SUPERMUTT, ARCADE ACTION ORBITER, STARSHIP ENTERPRISE, CYBER RATS, ARCADE ACTION, SLIPPERY SID	£5.00
CODENAME MATT II, VIEW TO KILL, FRIDAY 13th, TEST MATCH, BEAKY & THE EGG SNATCHERS, THEY PYRAMID	£5.95
IMAGINE PACK INC. STONKERS, PEDRO, ALCHEMIST, ZZOOM, ZIP ZAP, CONQUEST	£3.95
ON THE OCHE INC. SNOOKER, WORLD CUP FOOTBALL, OLYMPICS, GOLF, DERBY DAY	

UTILITIES

SPECDRUM	£25.00
QUICKSHOT II WITH M/S	£10.95
ALPHACOM PAPER 5 ROLLS	£ 8.95
CURRAH MICRO 5 LOTS	£ 2.95
COMPUTER TO T.V. LEADS	£ 1.50
MICRODRIVE CARTRIDGES	£ 1.99

p&p 75p all orders overseas – £1.00 per tape, £2.50 packs, £5.00 peripherals

LOGIC SALES LTD
6 Midgate · Peterborough · Cambs
24 hour ordering on 0733 313870

ROMANTIC ROBOT *present*

Nº1 MULTIPURPOSE INTERFACE FOR 48K & 128K SPECTRUM*

multiface one™ Version 48

New again! NEW ENHANCED VERSION AT THE OLD PRICE OF £39.95! WITH MORE HARDWARE & SOFTWARE THAN EVER BEFORE AND WITH MORE FEATURES THAN YOU FIND ON ANY OTHER INTERFACE

- Internal 8K ROM & 8K RAM enable FULLY AUTOMATIC saving of anything, anytime onto MICRODRIVES, WAFADRIVES, BETA, OPUS, KEMPSTON and TAPE with extra 8K at user's disposal and with an extensive MULTI TOOLKIT to study/modify/develop/programs.
- *Plus a joystick interface* (IN31, i.e. Kempston compatible) *and a switch* to make MULTIFACE ONE INVISIBLE.
- MULTIFACE ONE does not take any part of Spectrum RAM and does not need any additional software. Just push a button to freeze a program, save it and restart from the same point next time.
- MULTI TOOLKIT facilities allow to PEEK/POKE the entire 56K, show blocks of RAM with on-screen editing and decimal, hex or ASCII display, enable programmable jumps to anywhere within 56K, etc.
- Menu-driven with prompts and one-touch commands: 1) Push button 2) Select function: exit/return/save/tool/copy/jump 3) Input name 4) Save to: tape/cartridge/wafer/opus/beta/kempston (If using Kempston please specify when ordering)
- Extremely *powerful and efficient compressing* for fast re-loading & using minimal room on cartridges, tapes, wafers, disks.
- Option to SAVE a screen only
- COPY screen to printer (for printers interfaces accepting COPY command)
- Through extension bus for connecting other peripherals
- User-friendly, fully error-trapped, guaranteed, *simply magic*.
- *Programs saved with MULTIFACE can RUN independently*
- 128K Spectrum usable in 48K mode only on Version 48

Expand your Spectrum to 56K and stretch its capabilities even further with

multiface one ESSENTIAL SPECTRUM COMPANION

TRANS-EXPRESS

The software way of transferring Spectrum software. 4 utilities for m'drive, 2 for wafadrive, 2 for Opus Discovery. Invaluable for cartridge/disk maintenance & automatic transfers. Highly professional, user-friendly, versatile and comprehensive.
'An ingenious piece of software.' ***** Features ***** Performance PCN

TOP SOFTWARE AT BOTTOM PRICE

Wriggler

'One of the best games for any machine I've seen for very long time.' STAR GAME PCW
'Wriggler is a totally original game.' CRASH SMASH
'Great game, good music, well recommended.' What Micro
'The grahics and sound are excellent and the game is totally addictive.'
10/10 value & playability, 9/10 graphics & sound. C&V Games
'This is a great game.' CAS 'Well worth buying.' ZX C & Crash
'If you've any sense you'll order your copy now.' PCR 'A WINNER' PCW

!NOW FOR AMSTRAD! SPECTRUM version at a BUDGET price

MUSIC TYPEWRITER

Complete Spectrum music system for writing, editing, playing, printing & storing music. Its real notation graphics and ease of use make it an ideal educational tool. Great fun to play with & a lot to offer in serious use (transposing, dot matrix printout etc).

'A very professional piece of programming, easy to use, excellent screen presentation, error trapping & helpful prompts. If you're interested in music and own a computer then get it.' ZX Computing

MIDI compatible with MIDI MUSICIAN package
Now going for a song at £5·95 only

I enclose a cheque/PO for £.............. (UK & Europe orders please add £1 overseas £2)
or debit my [Access] [Visa] No [ ]
Name .... Card expiry ....
Address ....

Please send me (tick as applicable):

TRANS-EXPRESS cartridge	£9.95 ☐	MULTIFACE	£39.95 ☐
disk (Opus Discovery)	£9.95 ☐	tape (for m'drive)	£ 7.95 ☐
MUSIC TYPEWRITER	£5.95 ☐	wafer (Rotronics)	£ 9.95 ☐
		with supplement	£ 7.95 ☐
WRIGGLER Spectrum tape	£2.95 ☐	or cartridge	£ 5.95 ☐
WRIGGLER Amstrad tape	£7.95 ☐	or disk version	£ 11.95 ☐

ROMANTIC ROBOT 77 Dyne Road London NW6 7DR ☎ 24 hrs [Access] [Visa] orders 01·625 9463 ☎ ZX

OCTAGON SQUAD
Mastertronic
£1.99

This is one of those games that does not fit cosily into any of the usual categories applied to games, Mastertronic themselves describe it as an "icon driven arcade adventure" which just about covers all the available options.

However, to narrow it down a little, it is a maze game with strategy and adventure elements. I suppose that if one was being generous, the rather primitive moving graphics (I draw the line at calling it animation) would qualify it in the 'arcade' section.

If up to now I sound a little less than enthralled with the game, please do not be misled, it is a good program of its type and slowly grows on you. But first you have to overcome the cryptic and limited instructions, then get to grips with the screen display and icon system, then sort out who is who and what you are trying to do and how to do it.

After that it is a straightforward seek, collect and survive exercise.

Taking this program a step at a time pays off, I rushed straight from the menu to the game and missed some further instructions that are displayed automatically if you wait — they helped a little.

The main screen is bright and colourful, action is controlled by either five keys or joystick and is mastered after a bit of trial and error. The lower half containing a panel of twelve usable icons and three non usable ones and on the right a birds eye view of your character in the maze. At the top is a mini map and set of info windows.

You control a commander character and eight droids, one at a time by using the icon selection routine, and you move them around in a reactor maze to rescue 26 survivors of a heavy chem attack.

One of those games where patience and persistence is required to get the most from it, for the price and for fans of this kind of game I recommend it.







TRIVIAL PURSUIT
Domark
£14.95

The only potential problem with converting a phenomally successful board game like Trivial Pursuit to the home micro would seem to be tampering with a winning formula.

Few would have thought that the flaunting of irrelevant knowledge would exert such a hold over people but as trivial pursuit addicts know when the cheeses are down knowing whether frozen peas float or not assumes huge importance.

The Trivial Pursuit craze shows no signs of abating and Domark have produced a package faithful to the original but with several useful additions that add to the fun.

There are 3,000 questions, loaded in blocks ready to tax your memory for facts you would rather have forgotten before Trivial Pursuit came along. It's a two cassette pack. One being the game itself and the first batch of questions, the second an additional "Uniload" question pack which can be loaded into a Spectrum, Commodore or Amstrad.

One enhancement on the original is the inclusion of musical and graphics questions, so that for instance you can be confronted by the map of a country and asked to identify it.

Another inclusion is an animated question master called TP who bounds around the question screen in a number of guises. He's destined to become a "love him or hate him" figure. Some may feel he's a cute bonus while the purists will argue the original game got along just fine without him. Wisely Domark have added a cut off switch so you're not stuck with him.

You can fix a time limit for answering questions — anything from five seconds to nine minutes. A very useful feature for strategists is the inclusion of a score chart which keeps an up to the minute record of all the players performances. The breakdown is for each category of questions so when a player is going for the final winning question you can knobble him by confidently choosing his weakest subject.

Graphically the game is pleasing to the eye and the board layout is very clear. Most importantly perhaps it's very easy to play and you can find your way around the game and all you can do in next to no time.

Trivial Pursuit is bound to sell well on the strength of the board game and if you enjoyed that you won't be disappointed with its computer game counterpart.



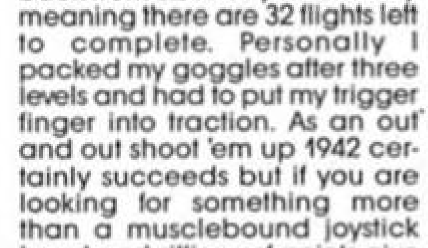
1942
Elite
£7.95

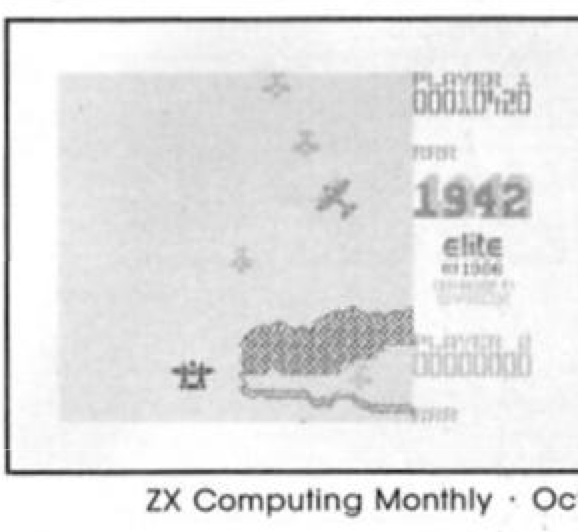
This is a conversion of a very successful arcade game but may not be an automatic hit unless there are sufficient shoot 'em up fans out there who prefer to do their blasting at home.

You take off in your fighter from the deck of a carrier and proceed across land and sea confronting waves of enemy planes. There are four phases on each level before you can land safely back on the carrier which really means just different permutations of planes coming at you. One phase has a huge bomber and bringing this down will bring you big bonus points but it's not necessary to destroy it to get onto the next phase.

Destroying certain planes will give you extra firepower or clear the screen of attacking planes for a moment. Apart from that there is little to divert you from the business of shooting the enemy out of the sky and amassing vast numbers of points. You are allowed a certain number of aerobatic rolls to get you out of trouble on each level and you have five lives in order to get as far as you can.

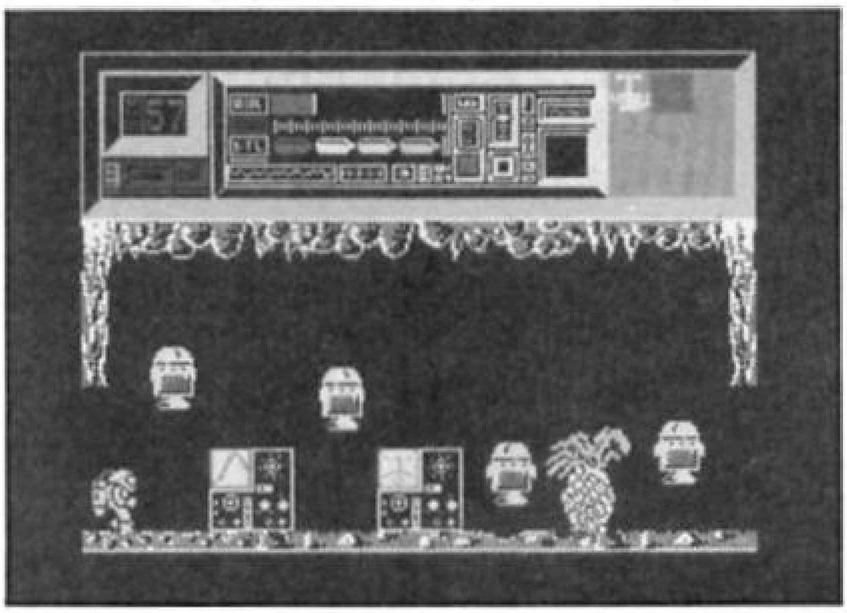
Landing back on the carrier gives you a thoroughly deserved 5000 point bonus before you take off again. Once back in the air it's the same old thing with just a few more planes and marginally more flak flying. Ominously the level gauge slips back from 32 to 31 presumably meaning there are 32 flights left to complete. Personally I packed my goggles after three levels and had to put my trigger finger into traction. As an out' and out shoot 'em up 1942 certainly succeeds but if you are looking for something more than a musclebound joystick hand and zillions of points give this one the flypast.





GAMES

UNIVERSAL HERO
Mastertronic
£1.99



'A totally nail-biting, multi-dimensional, hi-resolution Awesome Arcade Adventure!' And all that for just £1.99. Actually, Universal Hero isn't quite as wonderful as the tongue-in-cheek hype suggests, but it's still pretty nifty for a piece of budget software.

It seems that our Hero has just got seven seconds to save his planet, and in those seven seconds he's got to wander through a set of caverns and find the spare parts needed to repair his space shuttle, which will fly him to another planet where he has to collect the equipment needed to repair the out of control space freighter so that he can het back to earth in one piece.

Universal Hero is, as you might have guessed, a wander-'round-the-caves-collect-the-objects-and-avoid-the-monsters type game, and though it's a little old fashioned it is a highly polished and well designed example of the genre. And there's also a certain element of strategy thrown in as well, since some of the items that you have to locate can only be reached by solving puzzles. For instance, there is one cave entrance blocked by a pile of rubble and you've got to work out which objects can be used to start a rock slide and open up the entrance. Hero can carry up to nine objects in his pockets, so there's scope for some quite complex combinations of objects to be used.

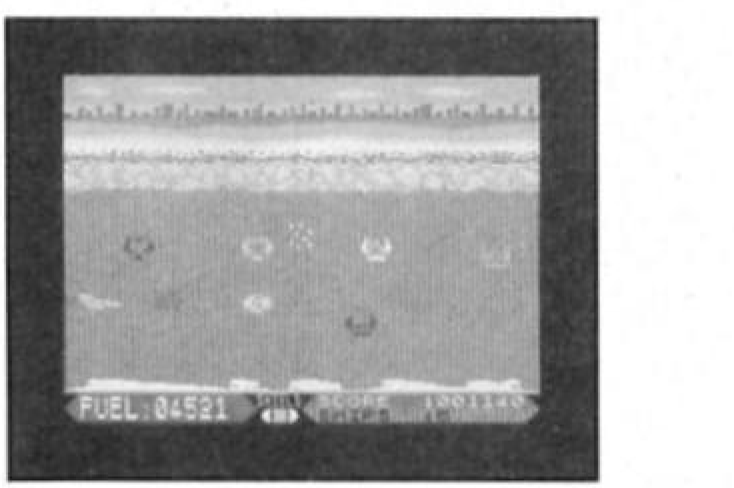
As he makes his way through the game, Hero uses up oxygen from four cylinders and collision with some of the nasties in the caves drains his air supply, so you've also got to go looking for spare oxygen supplies to keep you going. For some reason there are one or two monsters that just wipe out your entire supply in one go without any warning, and there's a lump of plutonium that has the same effect although again there's no warning that the object is actually plutonium. These things are a bit irritating as they make the game dependent on luck at some points, but once you've come across these traps once you can generally avoid them in future.

On the plus side there are some nice graphics and smooth animation, and the figure of Hero has a jaunty little walk as he strides through the caves. Thankfully there is a define keys option, but it would have been a good idea if the cassette inlay had given a bit more information about the game.

Universal Hero isn't a flawless game, but it does have its good points, and, as always, Mastertronic are still giving more expensive games a run for their money.



BUCCANEER
Firebird Silver 199 Range
£1.99



I wondered when this one would make its appearance! I reviewed three games from a company called Insight some months ago and since then two of the three have been repackaged and marketed by Firebird. Now this completes the set.

Described as a challenging arcade game in the classic style it means just that. A game on the lines of the classic "Space Invaders" type, wave after wave of nasties to be shot at while they try to shoot or crash into you. The action is different in that you are supposedly flying a Buccaneer aircraft and the aliens attack horizontally on an earth based scrolling background.

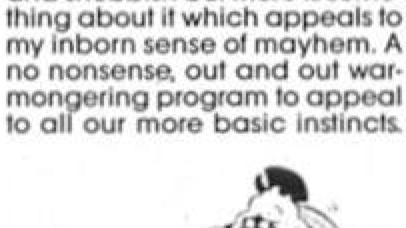
Just the kind of game I love to finish off a night of mind stretching arcventure reviewing. Sit down and zap away mindlessly. There are some extra features to alleviate the mind numbing effect, a docking sequence and a dodging sequence, but in the main it's a good old fashioned kill 'em all game.

At the original price of over seven pounds it was not worth having, but at the new £1.99 price it is the kind of program everyone should have, if only to compare how far the latest programs have progressed from the good old days.

Sophistication is out, but great fast and furious action survives with Buccaneer. Graphics and animation are average for this kind of game and sound bluzzes away merrily while colours clash cheerfully and death and destruction abound. Not one for pacifists.

You have five aircraft, or lives, and have to keep an eye on fuel and shield levels, controls are simply up, down and fire. Keyboard indentification and a full range of joystick options are included.

It's no good, I known I should hate this game and be all arty and snobbish but there is something about it which appeals to my inborn sense of mayhem. A no nonsense, out and out warmongering program to appeal to all our more basic instincts.



KANE
Mastertronic
£1.99

'A man's gotta do what a man's gotta do.' And in this case what you've gotta do is mosey on down to the old Spectrum and make peace with them thar injuns (well it makes a change from blowing their brains out doesn't it?).

You're the Sheriff of the town of Kane, and in order to make peace with the locals you have four tasks to complete. The first is to shoot down some birds so that you can trade with the indians for peace tokens. In this part of the game the screen shows a small cowboy figure at the bottom of the screen with mountains in the background and birds flying overhead. Using the keyboard or joystick you move a sight around the screen and attempt to shoot the birds down with your bow and arrow. You've got a limited supply of arrows, and with the birds being a moving target you've got to have a good sense of timing to get a decent amount of tokens. You need a token to move onto the next stage, and additional tokens give you extra lives in the later stages.

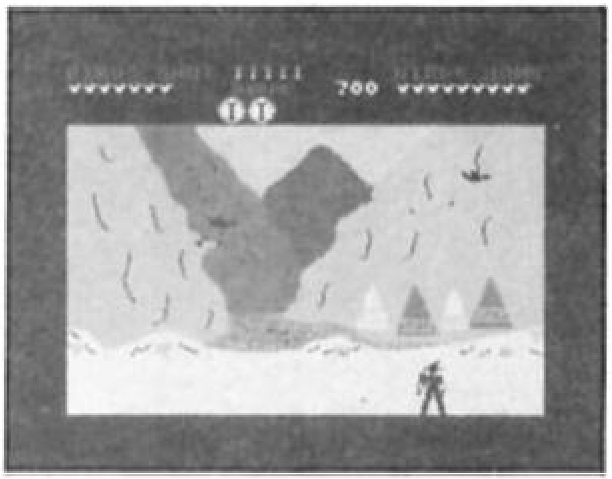
▶▶▶

Once you've demolished the local bird population you hop on your horse and ride across the prairie back to Kane. But along the way you've got to make sure that your old hoss doesn't go stumbling over any rocks or bushes or else you'll be out of the saddle and out of the game.

Once you've made it back to town you still don't get to rest as all the bad guys around have decided to take a pot shot at you. They're all hidden in doorways and around corners of the town's buildings, but every now and then they'll stick their heads up just long enough for you to get your sight on them and show who's boss.

After cleaning up the town it's back onto the horse for a race against the train as you try to overtake and stop it. Again you've got to jump over all the bushes and rocks, but this time there are small trees thrown in as well and some of the obstacles are so close together that you'll need a terrific sense of timing to make it past them all.

When I first saw Kane I thought that it looked pretty primitive — like a magazine listing in fact — but once I'd given it a couple of goes it turned out to be fairly good. It is very simple, but the difficulty of the tasks has been well pitched so that you can get started easily but still need to practice in order to get anywhere. Going through the same four stages in their fixed order all the time could get boring, but sensibly there's a practice mode which allows you to choose any stage of the game and one of three difficulty levels. This allows you to get a bit of practice on all the different parts of the game and then, when you think you're ready, you can try to tackle the whole thing in one go.



new standards in games programming, and if it looks old fashioned that's because it is, but it's enough fun to while away the odd rainy afternoon with and will probably appeal to younger computer owners who don't require state-of-the-art megagames.



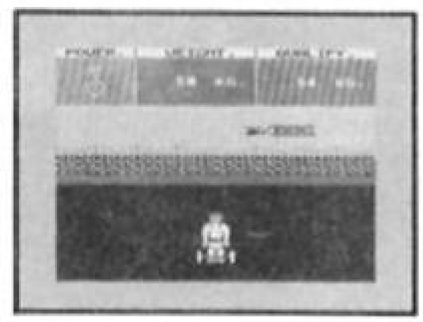
OLYMPIAD
Atlantis Software
£1.99

I'm at a loss as to how I can review this game constructively, other than to say that it simply isn't very good.

Olympiad is an attempt to produce a budget sports game along the lines of Daley Thompson's Super Test and includes five 'events' — Weight lifting, Canoeing, 200m Sprint, Skeet Shooting and Discus. For some of the events all that is involved is pressing a single button when the powermeter is at its height. The powermeter is a small clockface in the top left corner of the screen, around which a dial rotates constantly and when it reaches the 12 o'clock position (the highest energy level) you just stab the Q button. This is how you play the weight lifting and sprint sections and it doesn't give any of the excitement that the joystick/keyboard pounding style of Super Test provides.

The Canoeing section simply involves moving the canoe left/right across the screen in order to avoid the rocks that scroll upwards from the bottom of the screen. I completed this section without making a single mistake on my first attempt, so it can't really be described as much of a challenge.

The skeet shooting is the only part of the game that has any complexity to it — you actually get to use five controls for left/right/up/down and fire when positioning your guns sights.



The graphics aren't too bad, but they aren't good enough to make up for the lack of involvement in the games themselves. The program doesn't seem to accept joystick control and, when someone suggested that it might be simple Basic or compiled Basic, I tried to break the program just by using the BREAK key at which point a message flashed up saying 'Show Off! and then the program crashed. No points for user friendliness.'

I'm afraid the only thing that Olympiad has got going for it is the budget price, but even then it's not as good as some of the budget software from many other companies.



LAP OF THE GODS
Mastertronic
£1.99



Yet another interesting budget game from Mastertronic. In this, your task is to collect a number of gems in order to persuade the Gods of ZZarn to return you to your home. But before you can acquire the gems you will need to gain special powers by collecting the effigies that are scattered around the sixteen sets of chambers that make up the game's playing area.

As you gather together the various effigies you will be set upon by 'devils' that look remarkably like androids (and make me wonder who wrote the cassette notes, especially as they tell you the wrong keyboard controls). If the devils catch you they will take the effigy from you and return it to its resting place, but if you lose an effigy three times the gods slap your wrists by taking some of your powers and one of your lives too.

The sets of chambers are all separate, and the only connection between them is via the teleport chambers, but these will only work if you've collected the right effigies.

The screen display uses the outer section to display your lives left, crystals collected and so on, while the actual playing area is shown in a square inside this border. The graphics are relatively simple, with the passages and chambers drawn in quite large blocky graphics. These would be perfectly ok if the scrolling were a bit smoother, but as it is the jerky scrolling gave me quite an eyestrain, as did the rather distracting moving stripe all around the border of the playing area.

Still, ruptured eyeballs aside, LOTG is quite playable. It's one of those games that is simple and fairly repetitive but which draws you back for one more try to see if you can collect a complete set of effigies and move on to the next set of chambers.

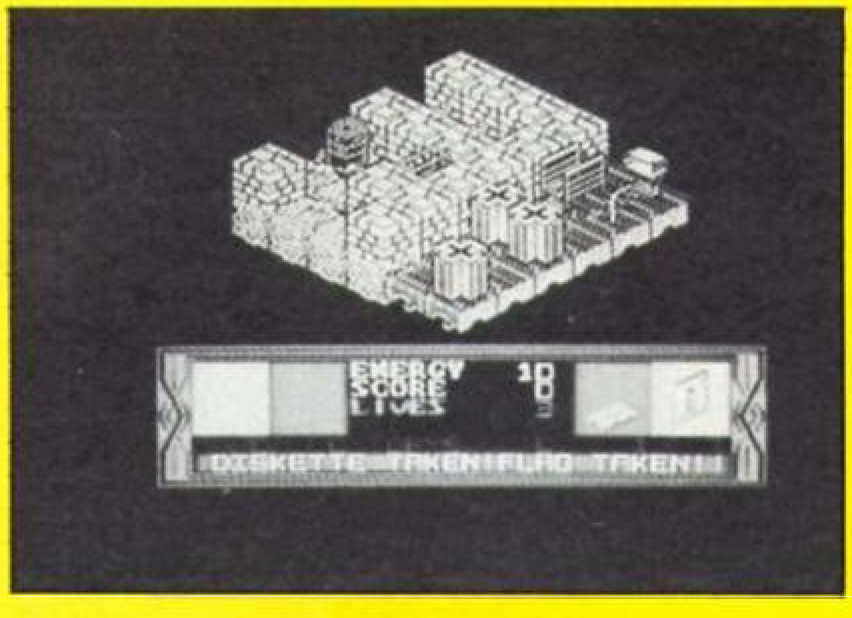
There is a 'fire' button, but this only pulls down a menu which displays the options and powers available to you (where would Mastertronic games be without a pull down menu, eh?). It would have helped the games playability if you were also able to fire at the 'devils' as they come at you. As it is, you're more or less at their mercy as they move faster than your own figure and can't really be easily avoided.

LOTG probably isn't the game for fast reaction acrade freaks, but if you've got a bit of patience, determination, and £1.99 to spare you might want to give it a go.



NAME THAT GAME!

COMPETITION



This is your chance to put a title to Mastertronic's Game With No Name.

Mastertronic are putting the final touches to a new Spectrum game and are giving ZX readers the opportunity to christen the release with a catchy imaginative title.

Not only will the winner get a kick out of seeing his title on the shelves of his local software emporium (and perhaps climbing up the charts) but he or she will also receive one of Mastertronic's new Magnum Joysticks, a Mastertronic T-shirt, posters, the five latest Mastertronic Spectrum releases and of course a copy of the Game With No Name.

There will also be 10 prizes of the Game With No Name and T-shirts for runners up.

The Game

The game has been written by Geoff Foley, author of Octagon Squad and Ticket To Ride. It features the exploits of a strange being apparently half foot, half head that is trying to escape from a planet surface of narrow pathways surrounded by water. In order to escape he must accomplish certain tasks like finding discs to place in disc drives, retrieving coins, locating energy packs and putting flags on flag poles.

As if all these tasks weren't enough there are also numerous twirling nasties stalking the pathways to destroy him.

Your turn

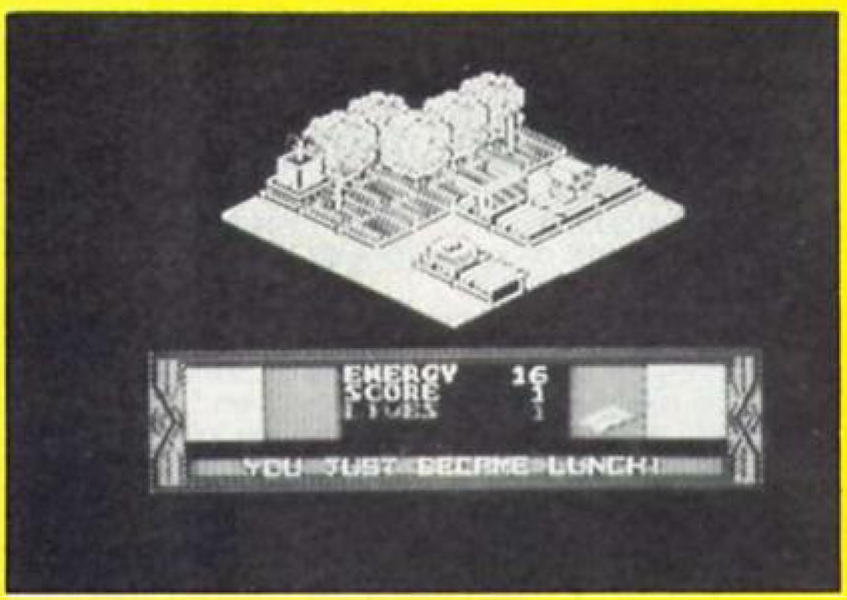
So now it's time for you to get pondering on the title. The prize will go to the originator of the most apt, original and above all catchy title for the game. Mastertronic have two provisional titles already and for any reader who inadvertantly guesses either of these titles there will be a special prize. (Using psychic powers to find these titles is strictly forbidden).

As Mastertronic are almost ready to release the game, the closing date for entries is closer than normal so get your snappy title in quickly and remember to write it on the outside of your entry envelope.

The closing date is October 24th 1986.

The competition is open to all ZX readers except employees of Argus Specialist Publications, Alabaster Pasmore and Mastertronic.

Send your entries to the Mastertronic Game With No Name Contest, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB.



This game is in search of a name



The Magnum Joystick

Mastertronic Game With No Name Contest

I name this game..

Name ....

Address ....

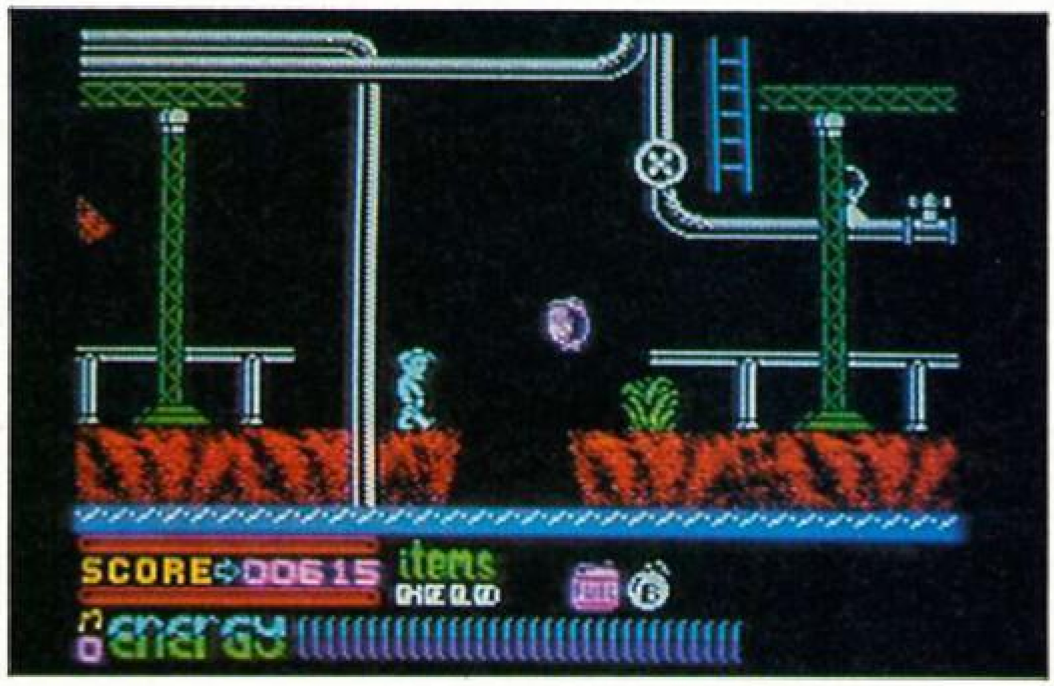
....

Send to Game With No Name Contest, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB. Please remember to put your title on the envelope.

Dynamite Dan is back to prove there's still life left in the platform game.

DYNAMITE DAN II

Dynamite Dan II
Mirrorsoft
£7.95



Just when you thought it was safe to come out of the cupboard, when you were quite sure that Doctor Blitzen and his evil bunch of cronies had finally been destroyed, and you had managed to float away to safety on the wings of a hot-air balloon, you must start all over again. Yes it's Dynamite Dan II.

When Dynamite Dan was first released, it had a lively aura about it, it was the type of game that you imagine would have been as much fun to program as it was to play. Thankfully, DD II seems to have lost none of the original charm and novelty.

What the passing of time has achieved is a better standard of game. All the definition is excellent, and unlike the original game, there is very little colour clash.

Together with many other people, I slightly miss the days when six hours spent jumping from ledge to ledge was six hours well spent. I miss the days when at least three new ledge games would be released every ten minutes, and I miss Jet Set Willy. Playing DD II brought all this back to me with a vengeance; the perfectly timed jumps, the agonising falls, they're all here, plus much more.

As you would expect, DD II does have an objective; you must travel around a total of eight islands, collecting a record on each, and inserting it into a jukebox. Having done this you must find some fuel for your trusty Zeppelin, and fly away. To help you in this seemingly never-ending task, there are a total of thirty two different objects, each with its own effect, varying from extra energy to immunity to bullets.



This may all sound very easy, but not surprisingly, life is made amazingly difficult by a stunning variety of baddies. They jump, spin, twist and drop around making life total and utter misery, and very nearly destroying your immaculately coiffured hair!

Perhaps the most impressive part of the game are the graphics. Varying from hi-tech, to jungle, they are brilliantly defined with obvious amounts of care. The characters too are animated with as much attention to detail as I have ever seen on the Spectrum.

If and when you get to the final island, you must blow up the last jukebox, and then make a desperate run for your Zep, within a matter of minutes. Only then will you have completed this 180+ screen masterpiece.

My only criticism would be the sound. Although there is a good attempt to get a tune and effects out of the Spectrum; the chip is so strained that it soon becomes annoying. However, even this cannot make DD II unenjoyable.

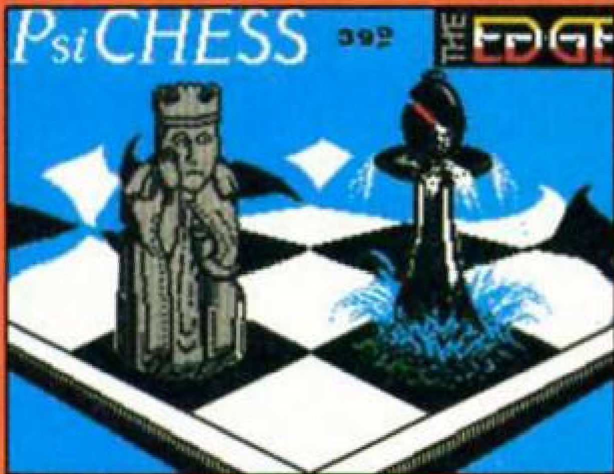
It is probably because there have been so few ledge games recently that DD II seemed so good, but I still believe that any arcade adventurer worth his salt will relish the thought of bouncing, jumping and swimming his way to success. Overall, DD II is a programming masterpiece, and on top of that it is real fun to play!



MONSTER HIT

GRANDMASTERS

Not one but two excellent chess programs have just been released for the Spectrum and both are a good alternative if Kasparov chickens out of your challenge match.





Psi Chess The Edge £9.95	**Basic Features**	Colossus 4 Chess CDS £9.95
2 Dimensional Board 3-D Board 3-D Isle of Lewis Pieces (Board can be rotated and seen from all four sides)	**Displays**	2 Dimensional Board 3-D Board
Seven 'Average' Levels. Response time varying from instantaneous to 3 minutes. Six 'Correspondence' levels. Blitz Chess — variable times. Balance of levels mode — the program attempts to match your response time. Beginner's level mode — the program handicaps itself to play a less than brilliant game.	**Levels**	Average mode — variable time limit providing hundreds of possible levels. Tournament mode — time controls on the number of moves made as in tournament play. All the Moves Mode — Blitz chess against the clock. Equity mode — the program keeps pace with your elapsed time clock. Infinite Mode — for puzzles or correspondence, the computer searches indefinitely for the best move.
Problems can be set up using an algebraic method or by joystick and icon to situate pieces. There is no "solve mate mode" but on some levels Mates can be solved in two or three moves.	**Problem mode**	Comprehensive mode for solving check mating problems. Can solve mates up to seven moves ahead also "Selfmates and helpmates".
Instant replay/takeback. Colour change options on board and pieces and border.	**Other features**	Instant replay/takeback. Colour changes options on board, pieces and border. Invisible Mode — enables you to play "blindfold chess" by making either black or white pieces (or both) invisible.
Joystick/keyboard.	**Control**	Joystick/keyboard.

Psi CHESS

MONSTER HIT

Spectrum Chessmen have never looked this good. Psi Chess goes for a visual as well as intellectual appeal.



Psi Chess features the sort of 3-D Chess graphics that previously have only been seen by Sinclair owners on QL Chess from Psion. Not content with depicting the usual Staunton pieces, programmer Steven Watson has also included a set of 3D Lewis chess pieces. These are medieval pieces depicting the King, Queen and other pieces as real figures. When seen on the Spectrum they look like refugees from an adventure game and add an extra element of novelty to the program.

Both 3-D sets are well defined with little confusion between pieces even in crowded areas of the board. Psi-Chess benefits from a full screen board display with only the time clocks and the present mode display as added detail.

Psi Chess is undeniably the most attractive looking chess program available for the Spectrum but, can it play? The answer is yes and although there are more powerful chess programs around it will certainly provide a stern test for even accomplished players.

Joystick chess

The operating system takes a little getting used to as it's based around three pivotal modes "Scoresheet, Position and Game". It can be a bit fiddly finding your way around and some basic commands take time to set up. For instance there is no instant restart to a game — if you want revenge you must press I for change mode, P and 1 together and the board resets. But to restart you must press I once more and then M and 1. If you are using a joystick this must also be selected again before a new game.

Joystick control can be a bit sensitive too. Unless you hit the piece that you want to move dead centre it can fail to "pick up" and you have to try again.

Psi Chess has a wide selection of levels and the lowest of these (the quickest) is instant response. This is so fast that it can take you several seconds scanning the board. I did manage to take a game off Psi Chess on Level 2 (with no take backs, honest!) but such is the seemingly omnipotent nature of computer games that I didn't quite believe it. A very smooth replay facility allowed me to wallow in the glory of victory by reshowing the game from start to finish.

This is a very accomplished package that comes complete with an informative 15 page manual. The emphasis on the graphic side can be seen from The Edge's intention to produce alternative 3-D chess sets that will be sold separately from Psi-Chess but can be incorporated by replacing the Lewis figures in the program's memory.

As exasperating as any good chess program should be, Psi Chess also allows you to savour the view as you slide towards ignominious defeat. A must for any Spectrum chess fanatic.



ZX COMPUTING MONTHLY







READERSHIP SURVEY

Win a free subscription!

This is your chance to tell us exactly what you think about ZX Computing Monthly. We want to know the features you enjoy and those you skip over. Are there subjects you would like to see covered in more depth or subjects you would like to see buried forever?

Your answers will help us to build up a picture of exactly the magazine you want to read.

All you have to do is fill in the questionnaire and fold into a postage paid envelope as detailed on the fourth page. We pay the postage so send off your answers as soon as possible because there are free subscriptions to ZX to be won for the senders of the first 10 surveys pulled out of the hat. Surveys must reach us by the closing date Friday, November 7th.

Your answers will be treated with the strictest confidence.

Bryan Ralph (Editor)

ZX COMPUTING READERSHIP SURVEY

1. Would you please tick the box against the statement that best describes how much of ZX you normally read:

Read through all features and reviews	☐ 1
Read through some features and reviews	☐ 2
Just read through the occasional features and reviews	☐ 3

2. Please tick the appropriate box that shows how you evaluate the following sections (Five is excellent, 1 is bad). If you would like to see extra features on a particular subject tick the appropriate 'more' box.

	1	2	3	4	5	More
News	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6
Games Reviews	☐ 7	☐ 8	☐ 9	☐ 10	☐ 11	☐ 12
Hardware	☐ 13	☐ 14	☐ 15	☐ 16	☐ 17	☐ 18
Utilities	☐ 19	☐ 20	☐ 21	☐ 22	☐ 23	☐ 24
Programming series	☐ 25	☐ 26	☐ 27	☐ 28	☐ 29	☐ 30
Mindplay	☐ 31	☐ 32	☐ 33	☐ 34	☐ 35	☐ 36
Competitions	☐ 37	☐ 38	☐ 39	☐ 40	☐ 41	☐ 42
Communications	☐ 43	☐ 44	☐ 45	☐ 46	☐ 47	☐ 48
Software House Profiles	☐ 49	☐ 50	☐ 51	☐ 52	☐ 53	☐ 54
Short Cuts	☐ 55	☐ 56	☐ 57	☐ 58	☐ 59	☐ 60
QL features	☐ 61	☐ 62	☐ 63	☐ 64	☐ 65	☐ 66
The 81 Page	☐ 67	☐ 68	☐ 69	☐ 70	☐ 71	☐ 72
Games Listings	☐ 73	☐ 74	☐ 75	☐ 76	☐ 77	☐ 78
Utility listings	☐ 79	☐ 80	☐ 81	☐ 82	☐ 83	☐ 84

What else would you like to see in ZX? ____________________

____

3. Which computer(s) do you own?

ZX 81	☐ 1	Spectrum	☐ 2
Spectrum +	☐ 3	Spectrum 128	☐ 4
QL	☐ 5	Other (please specify)	☐ 6

4. Do you own or have access to:

Printer	☐ 1	Modem	☐ 2
Discdrive	☐ 3	Microdrive	☐ 4

5. How much do you estimate your computer equipment is presently worth?

Under £220	☐ 1	From £220 to £300	☐ 2
From £300 to £400	☐ 3	From £400 to £500	☐ 4
Over £500	☐ 5		

6. How much do you estimate you have spent in total in the last twelve months on your computing activities?

£0-50	☐ 1	£51-100	☐ 2	£101-200	☐ 3
£201-500	☐ 4	Over £500	☐ 5		

7. If you own a Spectrum 48K are you considering buying the 128K version?

Yes ☐ 1 No ☐ 2

8. Do you use your computer for the following:

	All the time	More than half the time	Sometimes	Never
Original programmings	☐ 1	☐ 2	☐ 3	☐ 4
Typing in games listings	☐ 5	☐ 6	☐ 7	☐ 8
Typing in utility listings	☐ 9	☐ 10	☐ 11	☐ 12
Playing games	☐ 13	☐ 14	☐ 15	☐ 16
Educational uses	☐ 17	☐ 18	☐ 19	☐ 20
Business uses	☐ 21	☐ 22	☐ 23	☐ 24

9. Do software reviews influence your buying?

Yes ☐ 1 No ☐ 2

10. What types of games do you enjoy playing?

Arcade ☐ 1 Adventure ☐ 2 Strategy ☐ 3 Sports ☐ 4 None ☐ 5

11. Do you type in the games and utility listings in ZX?

	All	some	none
Games	☐ 1	☐ 2	☐ 3
Utilities	☐ 4	☐ 5	☐ 6

12. How many games do you buy in a year?

20 or more	☐ 1	15-20	☐ 2
10-15	☐ 3	5-10	☐ 4
Less than 5	☐ 5		

13. Does anyone else read your copy of ZX?

No, only me ☐ 1 1 or 2 others ☐ 2 3 or more ☐ 3

14. With regard to the advertisements in ZX, do you:

Read or look through most or nearly all of the ads	☐ 1
Read or look through some of the ads	☐ 2
Just read or look through the occasional ad.	☐ 2
Very rarely/never look at the ads.	☐ 4

15. Thinking specifically about the advertising content in ZX would you please rate the two types of advertising matter:— Display or Classified — in terms of usefulness (please tick one box against each type)

	Display	Classified
Very useful	☐ 1	☐ 2
Useful	☐ 3	☐ 4
Not very useful	☐ 5	☐ 6
Not at all useful	☐ 7	☐ 8

16. Have you ever ordered or bought products after reading an advertisement in ZX?

Regularly	☐ 1
Occasionally	☐ 2
Never	☐ 3

17. If the answer to question 16 is yes, what was the last item you purchased in this way, and what was its value?

....

18. Do you use the Advertisers Index?

Yes ☐1 No ☐2

19. Do you keep your copy of ZX for:

One month ☐1 Six months ☐2 A year or more ☐3

20. Which other computer magazines do you regularly buy?

Your Sinclair ☐1 Sinclair user ☐2

Crash ☐3 QL World ☐4

Other (please specify)

....

21. Did you read ZX when it was bi-monthly?

Yes ☐1 No ☐2

22. Since ZX went monthly do you think that the magazine is

Better ☐1 Worse ☐2 Same ☐3

Please state why ....

....

23. Are you aware of ZX's scheduled publication day?

Yes ☐1 No ☐2

24. If the answer to question 23 is yes, do you attempt to purchase the magazine on that day?

Yes ☐1 No ☐2

25. Do you normally obtain your copy by:

Chance purchase ☐1 Newsagent home delivery ☐2

Newsagent shop collection ☐3 Subscription ☐4

26, If you do not obtain your copy by subscription, is it due to one of the following?

Subscription too expensive ☐1

Not every issue required ☐2

Have subscribed previously, but lapsed ☐3

27. If you do not subscribe, from which type of newsagent do you most often obtain your copy?

High Street Shop ☐1 Estate shop ☐2

Travel point ☐3 Corner shop ☐4

28. Which of the following daily newspapers do you read?

Daily Mail ☐1 Daily Express ☐2

Daily Mirror ☐3 The Sun ☐4

Today ☐5 The Guardian ☐6

The Times ☐7 The Daily Telegraph ☐8

Financial Times ☐9 The Star ☐10

29. Which of the following Sunday Newspapers do you read?

The Sunday Times ☐1 The Sunday Telegraph ☐2

The Observer ☐3 Mail on Sunday ☐4

Sunday Mirror ☐5 Sunday Express ☐6

News of the World ☐7 The Sunday People ☐8

30. Do you read?

T.V.Times ☐1 Radio Times ☐2

31. Please tick the box which represents the annual total of your NET income (i.e. after tax, National Insurance and other deductions)

Pocket money ☐1 From £4,780 to £5,970 ☐2

From £5,980 to £7,470 ☐3 From £7,480 to £9,340 ☐4

From £9,350 to £11,680 ☐5 From £11,690 ☐6

32. It would be a great help to us if you would kindly answer the following questions. The answers will enable us to obtain a more positive image of our readership.

Under 15 ☐1 15-24 ☐2 25-34 ☐3 35-44 ☐4

45-54 ☐5 55-64 ☐6 65+ ☐7

33. Marital Status ....

Children ....

34. Do any of your children follow your interest in ZX Computing?

Yes ☐1 No ☐2

35. Name ....

Address ....

....

....

Sex: Male ☐1 Female ☐2

Occupation ....

36. Finally, have you any other comments to make about ZX?

....

....

....

TUCK INTO
THIRD FOLD

Postage will be paid by licensee

Do not affix Postage Stamps if posted in Gt Britain, Channel Islands, N Ireland or the Isle of Man

BUSINESS REPLY SERVICE
Licence No. WC 3970

2

ZX COMPUTING
Argus Specialist Publications Limited
No. 1, Golden Square,
LONDON W1R 3AB

FIRST FOLD

SECOND FOLD

THIRD FOLD

COLOSSUS 4 CHESS

The strongest chess program for the home micro finally reaches the Spectrum.

This program is definitely for the chess connoiseur — and the chess masochist. With an "opening book" of approximately 3000 positions and a move evaluaton system that examines on average 170 moves per second in the middlegame we are looking at a class program here and unless you are a class player you will be staring defeat in the face.

Of course there are features such as takeback and next/best move where the computer can be compelled to choose a second or third best move to soften the program's superiority (if you are unscrupulous), but even then it will give you a hard game.

The program is crammed with features and purchasers of the microdrive version get a bonus in the form of 34 games that can be loaded and analysed with the replay function.

Controls are simple to master and most major functions can be accessed quickly using the shift key. For instance if you want to deprive your computer opponent of time a useful command is SHIFT G which forces the computer to move immediately. There is also the Supervisor mode (SHIFT S) that stops the computer playing and allows you to make moves for both sides. It is also possible to switch sides in mid-game — a boon if you are being trounced or like hollow victories.

MONSTER HIT

3-D

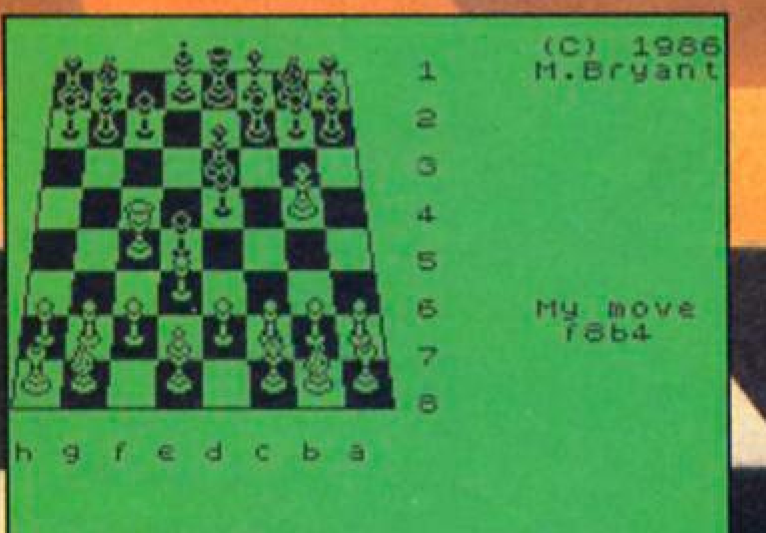
Graphically, Collosus 4 is more than adequate, though not stunning. The 3-D display although nice to have is not well enough defined to make it enjoyable to play with for any length of time. It can be confusing distinguishing which pieces are which and the display only takes up about half the screen. I found it more pleasing to play on the conventional two dimensional board where it's easier to keep track of the game.

If you want to watch Colossus in action against an opponent of equal stature (ie Colossus) there is a Play-self option that displays a game of computer-against-computer chess. By pressing the delete key you can step back into the action on either side while there is a pause between moves.

It is difficult to imagine a more complete chess program and more features than can be outlined here are described in an easy to follow 24 page booklet that comes with the game. The booklet also contains Colossus' past fixture list against other home computer chess games. Few managed to get more than the odd game in 16 and many suffered a humiliating whitewash. So it comes as no surprise that mere human opponents are cannon fodder for Colossus.

If you want a chess program to stretch you to the limit there is nothing to touch Colossus and I recommend it to anyone with an interest in Chess. A Grandmaster Monster Hit.





COLOSSUS Creator

Programmer Martin Bryant reflects on the grandmaster in the machine and its cheat-prone human opponent.



'I started writing chess programs when I was at University back in 1977. Then in 1982 I wrote White Knight for the BBC. I knew there could be a different approach so I decided to take the knowledge I'd aquired and write a better program.

Colossus I appeared first on the Apple in 1983 and since then there have been gradual improvements on newer versions. I usually take about six months improving the program and a year to write versions for other machines.

On improvements

'The catalyst to write a new version comes when someone brings out a new program to chalenge it. There is no clear cut way of improving a chess program although there is always something that can be improved. The problem is making the program as strong as possible within the time you have.

'There is no magic algorithm but the move evaluation function can be made more efficient. The values assigned to different capturing moves by the programmer give the program its specific style. It's important to make the computer ignore obviously stupid moves when deciding its next step. From any strategic position the program can generate all the possible moves in reply. This is 'game tree' theory and hopefully the best move comes from the branch you are looking at.

On fancy graphics

'Putting Colossus on the Spectrum caused a couple of problems with memory space but its playing strength is still the same as the Apple version. Economising is achieved by writing the code more efficiently and not getting carried away with the graphics. Obviously it would be nice to have fancy 3-D graphics as on the larger machines but playing strength is the program's prime resource and it would be silly to cut back on memory in that department. So in consequence the 3D pieces don't glide gracefully across the board.

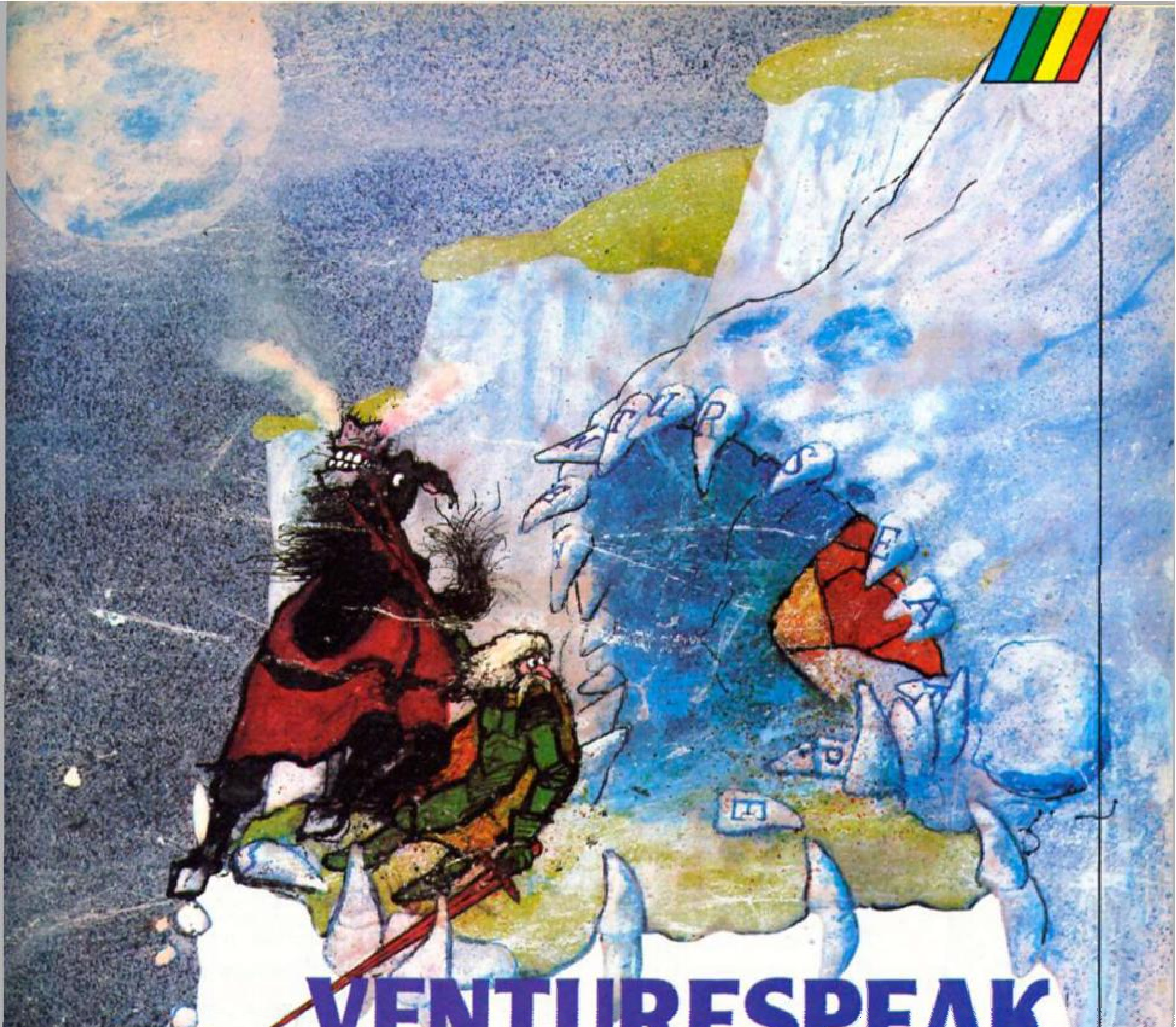
On cheating

'I don't play as well as I used to. Obviously I spend a lot of time playing against programs and it's very different because there is not the ego involved that comes out when playing a person.

'You can cheat very easily when you play the computer — take back moves, swop sides, and so on, and this can get you into bad habits. After all it's no good when you are playing a human opponent to ask him to give you your piece back.

'It depends on how you use it of course, if you keep taking moves back you are just kidding yourself. But what a chess program does is to provide a ready opponent any time who doesn't get tired.

'Chess programming is still of great interest to me and it never ceases to make me aware of the beauty of the game of chess. It's marvellous to be able to do a job of work that I really enjoy and I'll be playing any new programs that come out until a rival appears — then it's time to bring out Colossus 5 Chess.'



VENTURESPEAK

Alan Davis, recently returned from the Realms of Interaction sets out once more on a quest to produce a superior command analysis system.

It's often said that the quality of an adventure game is predominantly determined by the originality and imaginative scope of its plot, and really it would be hard to deny this. Virtuosity of programming, text compression, complex language analysis, and similar sophistications have been loudly trumpeted by many a software company, and equally loudly hailed by many a reviewer; yet these count for nothing if the heart of the game — its plot and atmosphere — is dull and uninspired. All this is true, but of course there are limits. It's equally possible for an adventure to be wonderfully imaginative in concept, but rendered almost unplayable by inordinately long response times, shabby on-screen presentation, or sheer "unfriendliness".

It's not difficult to find examples of this second category, even among recent commercial adventures: the grotesque "Lord of the Rings" comes most readily to mind, with its excruciatingly slow responses coupled to a keyboard input routine whose sluggishness makes accurate entry of commands a sore test of patience. Or how about those hordes of Quilled adventures, essentially rapid in response, but so often suffering from stereotyped presentation, unfriendly vocabulary, and the rigid restriction of the standard "verb/noun" input requirement.

Faced with this situation, the BASIC programmer with a yen to write adventures may well be put off for good. Even if he opts for the simple "verb/noun" input format, the need for a large, friendly vocabulary necessarily implies long delays in response whenever the program decodes the player's input and scans the vocabulary lists. Even then, if this limitation is accepted, there are many occasions when the "verb/noun" type of command analyser is inadequate — for the fact remains that you just can't

write an interactive "Hobbit"-style adventure without some capability for dealing with more complex commands.

Instant response

The obvious solution to the problem is to resort to machine code programming in order to deal with the command analysis aspects of adventure writing, since this has many advantages. First, program responses can be virtually instant, regardless of size of vocabulary. Second, complex and multiple commands (eg. PUT THE JEWEL IN THE CHEST AND LEAVE THE ROOM) can be dealt with, again at no cost in response time. And finally, extra facilities may be incorporated into the command analyser to permit, say, detailed editing facilities to improve "friendliness", or the option of "real time" operation in the completed adventure if required.

Such a solution in turn raises its own difficulties — not least that of interfacing the machine code routines with the rest of a predominantly BASIC program. We must also face the problem of (preferably) writing code which offers enough flexibility for us to use it for any adventure we may care to write. This, in a nutshell, is what the Venturespeak command analysis system has been designed for. Even if you have no understanding of machine code programming, you can use Venturespeak yourself to write sophisticated adventures which retain the flexibility and ease of BASIC, yet which are largely free from its speed limitations.

Venturespeak essentially consists of a single "package" which can be conveniently divided into 3 sections as follows:

1) A **Keyboard Input** routine which accepts typed commands up to 63 characters long, complete with single key-press facilities for deleting/editing commands, and a real-time option.
2) A **Parser** routine which decodes complex commands containing up to three verbs and four nouns, such as "ASK SAM TO PUT THE COIN IN THE PURSE", as well as permitting several commands to be strung together in sequence using commas, full stops, or "AND".
3) The menu-driven **Venturespeak Editor** which offers comprehensive and easy-to-use facilities for adding, deleting, or changing vocabulary. It is capable of generating a single block of code which incorporates both keyboard and parsing routines together with the vocabulary needed for any particular adventure.

Listing 1

```
*HISOFT GENS3M2 ASSEMBLER*
      ZX SPECTRUM

Copyright (C) HISOFT 1983,4
All rights reserved

Pass 1 errors: 00

                      *D+
             *C-
60000                 ORG   60000
60000        KEYBD    LD    A,0
60002                 LD    (23658),A
60005                 CALL  CLEAR
60008                 CALL  PROMPT
60011        RESET    XOR   A
60012                 LD    (23672),A
60015                 LD    (23673),A
60018                 LD    (23674),A
60021                 LD    (COUNT),A
60024        WAIT     XOR   A
60025                 LD    (23560),A
60028        WAIT2    LD    A,(23673)
60031                 CP    2
60033                 JP    NC,TIMEUP
60036        CONT     LD    A,(23560)
60039                 CP    0
60041                 JP    Z,WAIT2
60044                 CP    6
60046                 JP    Z,REPEAT
60049                 CP    13
60051                 CALL  Z,CHECKN
60054                 CP    0
60056                 JP    Z,WAIT
60059                 CP    12
60061                 CALL  Z,CHECKN
60064                 CP    0
60066                 JP    Z,WAIT
60069                 CP    123
60071                 JP    NC,WAIT
60074                 LD    (CHR),A
60077                 CP    13
60079                 JP    Z,ENTER
60082                 CP    12
60084                 JP    Z,DELETE
60087                 LD    A,(CHR)
60090                 CP    7
60092                 JP    Z,WIPE
60095                 CP    32
60097                 JP    C,WAIT
60100                 LD    A,(COUNT)
60103                 CP    63
60105                 JP    Z,WAIT
60108                 LD    HL,START
60111                 LD    D,0
60113                 LD    E,A
60114                 LD    A,(CHR)
60117                 ADD   HL,DE
60118                 LD    (HL),A
60119                 INC   DE
60120                 CALL  CURSOR
60123                 LD    A,E
60124                 LD    (COUNT),A
60127                 JP    WAIT
60130        CLEAR    XOR   A
60131                 CALL  #1601
60134                 CALL  SETPR
60137                 LD    B,64
60139        SCAN     LD    A,32
60141                 PUSH  BC
60142                 RST   16
60143                 POP   BC
60144                 DEC   B
60145                 LD    A,B
60146                 CP    0
60148                 JP    NZ,SCAN
60151                 CALL  SETPR
60154                 RET
60155        CHECKN   LD    (CHR),A
60158                 LD    A,(COUNT)
60161                 CP    0
60163                 JP    Z,RESETA
60166                 LD    A,(CHR)
60169        RESETA   RET
60170        ENTER    CALL  CLEAR
60173                 LD    A,1
60175                 LD    (TIME),A
60178                 LD    A,(COUNT)
60181                 LD    (CT2),A
60184                 RET
60185        WIPE     XOR   A
60186                 LD    (COUNT),A
60189                 CALL  CLEAR
60192                 CALL  PROMPT
60195                 JP    WAIT
60198        SETPR    LD    A,#16
60200                 RST   16
60201                 XOR   A
60202                 RST   16
60203                 XOR   A
60204                 RST   16
60205                 RET
60206        PROMPT   LD    A,62
60208                 RST   16
60209                 RET
60210        TIMEUP   LD    A,(COUNT)
60213                 CP    0
60215                 JP    NZ,CONT
60218                 CALL  QUIT
60221                 RET
60222        CURSOR   PUSH  DE
60223                 LD    A,17
60225                 RST   16
60226                 LD    A,6
60228                 RST   16
60229                 LD    A,(CHR)
60232                 RST   16
60233                 POP   DE
60234                 RET
60235        SPACE    LD    A,17
60237                 RST   16
60238                 LD    A,7
60240                 RST   16
60241                 LD    A,32
60243                 RST   16
60244                 RET
60245        DELETE   LD    A,(COUNT)
60248                 CP    31
60250                 JP    Z,TLINE
60253                 LD    A,8
60255                 RST   16
60256        HERE     CALL  SPACE
60259                 LD    A,8
60261                 RST   16
60262                 LD    A,(COUNT)
60265                 DEC   A
60266                 LD    (COUNT),A
60269                 JP    WAIT
60272        TLINE    LD    A,#16
60274                 RST   16
60275                 XOR   A
60276                 RST   16
60277                 LD    A,31
60279                 RST   16
60280                 JP    HERE
60283        REPEAT   LD    A,(COUNT)
60286                 CP    0
60288                 JP    NZ,WAIT
60291                 CALL  CLEAR
60294                 CALL  PROMPT
60297                 LD    DE,START
60300                 LD    A,(CT2)
60303                 LD    (COUNT),A
60306                 LD    B,0
60308                 LD    C,A
60309                 CALL  #203C
60312                 JP    WAIT
60315        COUNT    DEFB  0
60316        CT2      DEFB  0
60317        CHR      DEFB  0
60318        START    DEFS  65
60383        TIME     DEFB  0
60384        QUIT     LD    (TIME),A
60387                 CALL  CLEAR
60390                 RET

Pass 2 errors: 00

CHECKN  EAFB        CHR     EB9D
CLEAR   EAE2        CONT    EA84
COUNT   EB9B        CT2     EB9C
CURSOR  EB3E        DELETE  EB55
ENTER   EB0A        HERE    EB60
KEYBD   EA60        PROMPT  EB2E
QUIT    EBE0        REPEAT  EB7B
RESET   EA6B        RESETA  EB09
SCAN    EAEB        SETPR   EB26
SPACE   EB4B        START   EB9E
TIME    EBDF        TIMEUP  EB32
TLINE   EB70        WAIT    EA78
WAIT2   EA7C        WIPE    EB19

Table used:   321 from   345
```

Assembly points

This month we'll be dealing with the input routine, and the other aspects of Venturespeak will be tackled in succeeding articles, but before we begin I'd like to make a strong recommendation that you equip yourself with a decent assembler program (if you don't have one already). You *can* poke in the machine code routines from BASIC of course — and I'll provide decimal dumps for those who wish to do so — but we'll be dealing with over 1K of machine code altogether, and the scope for errors when typing in the long lists of numbers is considerable, even with checksums. An assembler is well worth the investment — I use Hisoft's "DEVPAC" and find it excellent.

Now that the hors-d'oeuvres are out of the way, we can get on with this month's main course. Before you start, type CLEAR 59999 to lower RAMTOP so that our code can be stored safely at address 60000 and above. Incidentally, I've chosen this address so that the Venturespeak routines will be compatible with those from the previous "Realms of Interaction"

Listing 2

ADDR.	BYTES					CHECK SUM
60000	62	0	50	106	92	310
60005	205	226	234	205	46	916
60010	235	175	50	120	92	672
60015	50	121	92	50	122	435
60020	92	50	155	235	175	707
60025	50	8	92	58	121	329
60030	92	254	2	210	50	608
60035	235	58	8	92	254	647
60040	0	202	124	234	254	814
60045	6	202	123	235	254	820
60050	13	204	251	234	254	956
60055	0	202	120	234	254	810
60060	12	204	251	234	254	955
60065	0	202	120	234	254	810
60070	123	210	120	234	50	737
60075	157	235	254	13	202	861
60080	10	235	254	12	202	713
60085	85	235	58	157	235	770
60090	254	7	202	25	235	723
60095	254	32	218	120	234	858
60100	58	155	235	254	63	765
60105	202	120	234	33	158	747
60110	235	22	0	95	58	410
60115	157	235	25	119	19	555
60120	205	62	235	123	50	675
60125	155	235	195	120	234	939
60130	175	205	1	22	205	608
60135	38	235	6	64	62	405
60140	32	197	215	193	5	642
60145	120	254	0	194	235	803
60150	234	205	38	235	201	913
60155	50	157	235	58	155	655
60160	235	254	0	202	9	700
60165	235	58	157	235	201	886
60170	205	226	234	62	1	728
60175	50	223	235	58	155	721
60180	235	50	156	235	201	877
60185	175	50	155	235	205	820
60190	226	234	205	46	235	946
60195	195	120	234	62	22	633
60200	215	175	215	175	215	995
60205	201	62	62	215	201	741
60210	58	155	235	254	0	702
60215	194	132	234	205	224	989
60220	235	201	213	62	17	728
60225	215	62	6	215	58	556
60230	157	235	215	209	201	1017
60235	62	17	215	62	7	363
60240	215	62	32	215	201	725
60245	58	155	235	254	31	733
60250	202	112	235	62	8	619
60255	215	205	75	235	62	792
60260	8	215	58	155	235	671
60265	61	50	155	235	195	696
60270	120	234	62	22	215	653
60275	175	215	62	31	215	698
60280	195	96	235	58	155	739
60285	235	254	0	194	120	803
60290	234	205	226	234	205	1104
60295	46	235	17	158	235	691
60300	58	156	235	50	155	654
60305	235	6	0	79	205	525
60310	60	32	195	120	234	641
60315	2	2	13	75	76	168
60320	255	32	83	80	65	515
60325	67	69	255	65	78	534
60330	68	32	71	79	32	282
60335	73	78	255	84	72	562
60340	69	32	74	69	87	331
60345	69	76	32	73	78	328
60350	32	84	72	69	32	289
60355	67	72	69	83	84	375
60360	255	84	72	255	47	713
60365	63	58	61	43	60	285
60370	62	59	59	59	34	273
60375	34	39	39	40	41	193
60380	38	0	0	1	50	89
60385	223	235	205	226	234	1123
60390	201	0	0	0	0	201

series, should you wish to combine them. (The code is *not* relocatable).

Listing 1 gives the assembly language program for the keyboard input routine. The code is organised at address 60000, and is 391 bytes long. For those assemblerless individuals, Listing 2 provides the (decimal) bytes you'll need in conjunction with the BASIC loader giving in Listing 3. Type in Listing 3, and then RUN. Just follow the prompts the program gives you, entering the individual bytes from Listing 2 one at a time, including the checksum after every fifth byte. If a mistake is found, the program will ask you to enter the whole of the offending line of 5 bytes again. The screen display which builds up should be identical to the layout of Listing 2, and this should help you to check everything as you go along. Once the whole thing has been entered without errors, the program will save your code to tape as "KEYBOARD" CODE 60000, 391.

What, then, does this routine do when you call it with LET m = USR 60000? Well, basically it prints a small "prompt" symbol at the bottom of the screen and waits for you to start typing a command. Each character you type is stored as a single byte above RAMTOP, the bytes being stored sequentially from address 60318 (START) onwards up to a maximum of 63 characters. When you finally press ENTER, the routine does an immediate return to BASIC, with the total length of your typed command held at address 60315 (COUNT). All is then ready for the parser (which of course we don't yet have) to take over and analyse your command. However, this is by no means all that the routine can do for you. The following facilities are available:

1) CAPS SHIFT/0 acts just like the usual Spectrum "DELETE" key, erasing the last character typed.
2) CAPS SHIFT/1 deletes everything you've typed so far.
3) CAPS SHIFT/2 enables you to recall the whole of the last command you entered, either so that you can enter it again unchanged, or for editing using the "DELETE" key.

Zero time

The routine also has a built-in real time facility which operates as follows: if nothing has been typed, then after about 10 seconds a return to BASIC occurs with the address 60383 (TIME) holding the value zero. If on the other hand a return to BASIC has occurred due to the player entering a command, the address 60383 holds the value 1.

Listing 3

```
   1 REM ++ M/C LOADER
   2 REM
   3 REM
  10 PRINT "ADDR.";TAB 13;"BYTES
";TAB 26;"CHECK"'TAB 28;"SUM"''
  20 FOR I=60000 TO 60390 STEP 5
  30 PRINT I;
  40 LET S=0
  50 LET N=0
  60 INPUT "NEXT BYTE?"'X: IF X>
255 THEN GO TO 60
  70 POKE I+N,X: LET S=S+X: PRIN
T TAB 6+4*N;" " AND X<100;" " AN
D X<10;X;
  80 LET N=N+1: IF N<5 THEN GO T
O 60
  90 INPUT FLASH 1;"CHECKSUM FOR
 THIS LINE?"'C: PRINT TAB 27;" "
 AND S<10;" " AND S<100;" " AND
S<1000;S: IF C<>S THEN PRINT "ER
ROR! ENTER THIS LINE AGAIN.": BE
EP 1,30: GO TO 30
 100 NEXT I
 110 SAVE "KEYBOARD"CODE 60000,3
91
```

Listing 4

```
   1 REM    KEYBOARD TEST ROUTINE
   2 REM
   3 REM
   5 LET input=60000: LET start=
60318: LET count=60315: LET time
=60383: LET scroll=3582
  10 CLS
  20 LET m=USR input
  30 IF NOT PEEK time THEN PRINT
 AT 20,0;"Time passes.": GO SUB
100: GO TO 20
  40 PRINT AT 20,0;: FOR i=start
 TO start-1+PEEK count: PRINT CH
R$ PEEK I;: NEXT I
  50 GO SUB 100: GO TO 20
 100 LET m=USR scroll: IF PEEK c
ount>32 THEN LET m=USR scroll
 110 RETURN
```

This means that on exit from the routine we can test PEEK 60383, and execute appropriate subroutines (say for independent action of characters) if that memory location contains zero. If on the other hand it contains the value 1, we know that a command analysis must be carried out first.

If you don't want your adventure to run in real time, this feature can be suppresed by amending the code with the following POKEs: POKE 60033,0: POKE 60034,0: POKE 60035,0. (You might like to save two versions of the code — one with real time, and one without). Alternatively, if you want to keep the real time feature but slow down the pace of things, try poking different values into 60032. POKE 60032, 4 will give exits to BASIC every 20 seconds or so, for instance.

Still, trying all these things for yourself is much more fun than reading about them, and Listing 4 will enable you to check the routine out. First CLEAR 59999, load in the bytes, type in the lines of BASIC from Listing 4, and then RUN. When you see the prompt at screen bottom, type in any old command (note that CAPS LOCK is set automatically) and press ENTER. If all is well the lower screen will clear, and your command will be exactly reproduced in the upper screen (confirming that it has been stored correctly in memory).

Try pressing CAPS SHIFT/2 to check that the "recall last command" facility is working correctly, and in addition check the delete functions. It's clear, I hope, what the benefits of these facilities are: suppose your last command was "ASK SAM TO ATTACK THE ORC", and that on your next turn you'd like Sam to beat the daylights out of a troll. All you need do is recall your previous command using CS/2, delete the word "ORC", and add the word "TROLL" before pressing ENTER.

All OK? Fine. Make yourself a well-earned cup of coffee, for you now have about one third of the complete Venturespeak system safely stashed away on tape. See you next month, with the second instalment. . .

MACRO-LIFE

Toni Baker plumbs the mysteries of creation with an intriguing program for the Spectrum. Just like real life, Macro Life is what *you* make it.

This is what you might call an artistic program. Not an artist's program, you understand, but an artistic program, for it is the program itself which creates all the patterns — all you have to do is start it off.

The program is called Macro-Life, and is essentially a version of Life, a program which is supposed to represent the birth/growth/death cycle of a colony of cells living on a grid. It produces rather fascinating results. Before your eyes you see a constantly evolving pattern — always changing, always growing — almost as if it were alive. Who knows — perhaps it is.

Life was invented in 1970 by a man called John Conway of Cambridge University, and I believe that the Tate Gallery really ought to have a version of LIFE running at all times, because although it is intended to represent the growth of cells which follow hard and fast mathematical rules, in practice it becomes a rather effective pattern generating algorithm.

The rules of Life (or indeed Macro-Life) are very simple. A grid is initialised by having a number of its available squares filled with cells (A square may either contain a cell, or it may be empty. Only one cell at most is allowed per square). This initial configuration is called generation zero.

Successive generations are worked out by a fairly simple-to-understand process. Every square on the grid is surrounded by eight other squares. These neighbouring squares may or may not contain cells. What you do is count how many of the neighbouring squares contain cells. Every cell which has precisely either two or three neighbours will survive to the next generation, and all other cells will die. In addition, new cells will be born in every empty square which has precisely three neighbouring cells. Although these rules are simple, the algorithm produces surprisingly elegant results.

A better life

I have seen quite a few versions of Life in my time. I have seen a small, eight by eight version; I have seen a couple of sixteen by sixteen versions (this is the usual size); I have also seen a twenty-four by twenty-four version, which was also quite interesting. The largest version I have seen so far is a thirty-two by twenty-four square version which used the whole of the Spectrum's screen, with one character square for each square of the grid.

This version of Life is quite over the top. it uses a grid of 256 by 192, with a single pixel for each grid-square. It is the largest version of Life which could conceivably fit onto the Spectrum's screen. The patterns which it produces are dramatically different from those produced by smaller versions — they are more intricate and delicate, rather like a close up of a snowflake (except that the symmetry here is four-fold, not six-fold).

When I ran the program for the first time, after having written it and got it working, I was quite stunned by the intricacy and beauty of the designs it was producing. I felt that since the overall effect was markedly different from the smaller versions of Life, then it needed a different name. Such sanctimonious titles as "Mega-Life" and "Super-Life" came to mind, but it was John Conway's original idea which gave me the idea for the title I have used. You see, Life is supposed to show cells multiplying and dying. Living cells are, generally speaking, microscopic. In my program, however, individual cells may not even be discernable (depending on the quality of TV picture), so that the overall patterns are determined by large numbers of cells collectively — cell structures. Large cell structures are macroscopic (rather than microscopic), and are a result of the underlying microscopic cells. For this reason, and also because of the double meaning of the word "macro" (it also means a machine code procedure which may be duplicated several times in a program) I decided to call the program Macro-Life.

The machine code part of this program does NOT create the initial configuration (generation zero). Instead, I use BASIC for this purpose. Lines 1000 and 1010 integrate the machine code into the BASIC. Line 1000 generates the next pattern, while 1010 is just a repeat loop, which also enables the user to break out by pressing BREAK (or CAPS SHIFT/SPACE) in between generations. All BASIC lines between 1 and 999 may be used to create generation zero. I have provided seven examples of small BASIC programs which when merged with lines 1000 and 1010 will provide beautiful examples of the kind of thing that Macro-Life is capable of.

You can, of course, write your own BASIC programs to create the initial configuration, and indeed this was my deliberate intention, and the reason that I left the construction of this initial pattern to BASIC.

The machine code begins at address 8000 (hex) and is less than two hundred bytes long. The machine code runs from the label NEXT_GEN — this is the address referred to in BASIC line 1000. In addition, whilst running. The program uses addresses C000 to D7FF to store a back-up copy of the screen. The back-up copy contains the "last generation", while the current generation is being calculated directly onto the screen.

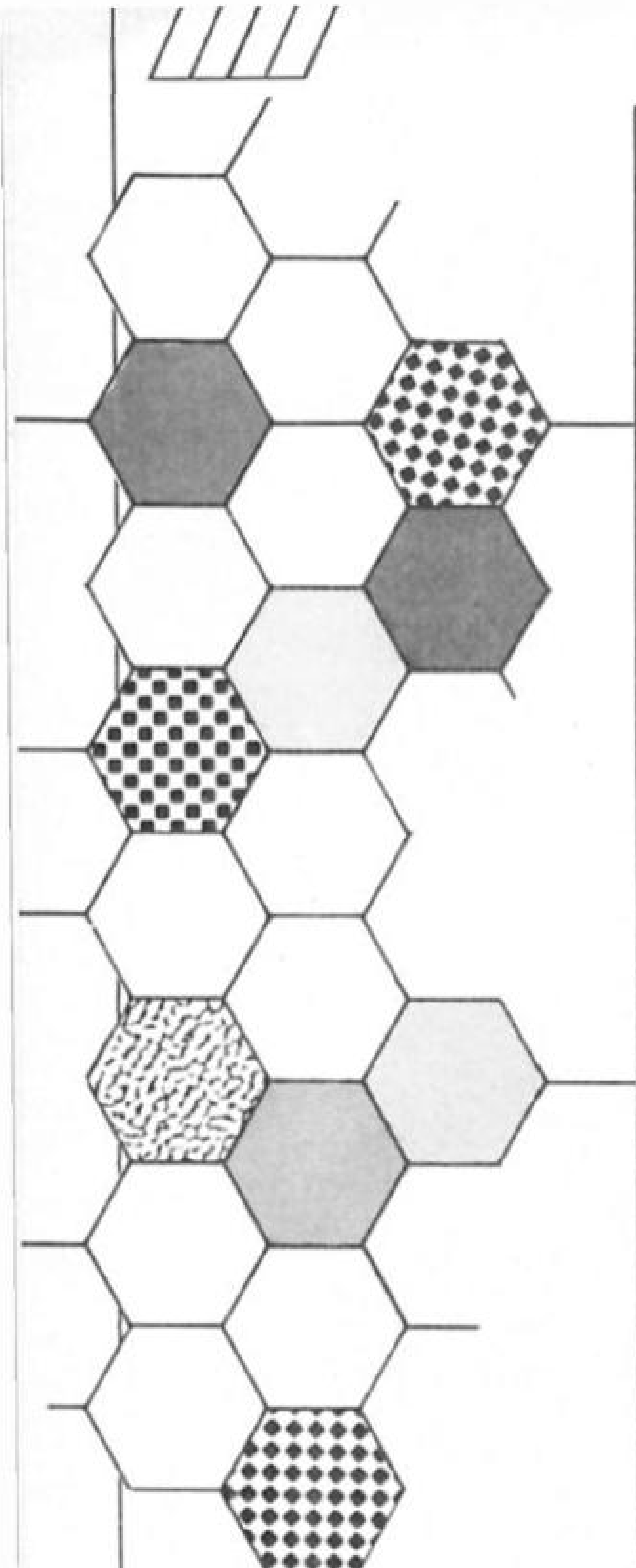
Cell blocks

I've also discovered that interesting effects occur if you place new live cells onto the grid between successive generations. Try deleting all lines before 1000 and adding a new line 1005 CIRCLE 128,80,80.

The grid used for this program exists in a strange warped space continuum, whereby the left hand edge is joined to the right hand edge (and vice versa), and the top is joined to the bottom (and vice versa). Technically this represents a doughnut shaped universe. You might try changing this feature — you can do this quite easily in BASIC by killing all cells around the edges. Adding line 1005 INVERSE 1: PLOT 0,0: DRAW 255,0: DRAW 0,175: DRAW −255,0: DRAW 0,−175: INVERSE 0 will do the trick. You can try out this, as well as many other experiments. Use your imagination, sit back and watch the speccy go to work.

```
                    ORG 8000
25      UP          DEC H               Up one pixel (assuming within chr square).
7C                  LD A,H              A:= high part of address.
F6F8                OR F8               Use only bits 2, 1 and 0.
3C                  INC A
2041                JR NZ,EXIT          Jump if within character square.
7D                  LD A,L              A:= low part of address.
D620                SUB 20
6F                  LD L,A              HL:= pix addr, assuming crossed from one
                                            screen-third to another.
7C                  LD A,H              A:= high part of address.
3805                JR C,TEST_TOP       Jump if crossed from one third to another.
C608                ADD A,08
67                  LD H,A              HL:= pixel address (all remaining cases).
1835                JR EXIT             Jump to exit.
FEBF    TEST_TOP    CP BF
2031                JR NZ,EXIT          Jump unless off top of screen.
26D7                LD H,D7             Point HL to bottom line of screen. Note bit
182D                JR EXIT                 7 of H set for all screen addresses here.

24      DOWN        INC H               Down one pixel (assuming within chr square).
7C                  LD A,H              A:= high part of address.
E607                AND 07              Use only bits 2, 1 and 0.
2027                JR NZ,EXIT          Jump if within character square.
7D                  LD A,L              A:= low part of address.
C620                ADD A,20
6F                  LD L,A              HL:= pix addr, assuming crossed from one
                                            screen-third to another.
7C                  LD A,H              A:= high part of address.
3805                JR C,TEST_BOTTOM    Jump if crossed from one third to another.
D608                SUB 8
67                  LD H,A              HL:= pixel address (all remaining cases).
181B                JR EXIT             Jump to exit.
FED8    TEST_BOTTOM CP D8
2017                JR NZ,EXIT          Jump unless off bottom of screen.
26C0                LD H,C0             Point HL to top line of screen.
1813                JR EXIT             Jump to exit.
CB01    LEFT        RLC C               Left one pixel (assuming within chr square).
300F                JR NC,EXIT          Jump if within chr square.
7D                  LD A,L              A:= low part of address.
3D                  DEC A               Decrement (to move left).
1806                JR L_R
CB09    RIGHT       RRC C               Right one pixel (assuming within chr square).
3007                JR NC,EXIT          Jump if within character square.
7D      RIGHT_1     LD A,L              A:= low part of address.
3C                  INC A               Increment (to move right).
AD      L_R         XOR L               Combine A and L by taking bits 7, 6 and 5
E61F                AND 1F                  of L, and bits 4, 3, 2, 1 and 0 of A.
AD                  XOR L                   This has the effect of "joining" the
6F                  LD L,A                  left and right hand edges of the screen.
79      EXIT        LD A,C              A:= pixel position within square.
A6                  AND (HL)            Test the specified screen pixel.
C8                  RET Z               Return if the pixel is PAPER colour (reset).
04                  INC B               Otherwise increment count in B register.
C9                  RET
                    ORG 804D
210040  NEXTGEN     LD HL,4000          Point HL to screen.
1100C0              LD DE,C000          Point DE to screen copy.
010018              LD BC,1800          BC:= number of bytes in screen.
D5                  PUSH DE             Stack start address of screen copy.
EDB0                LDIR                Make copy of screen.
E1                  POP HL              HL points to first byte of screen copy.
0E80                LD C,80             C has leftmost pixel set, thus HL and C
                                            together point to first pixel in scr copy.
0600    LOOP        LD B,00             Initialise cell count as zero.
```



```
C5                  PUSH BC              Stack pixel position.
E5                  PUSH HL              Stack screen-copy address.
CD0080              CALL 8000,UP         Count, in B, the number of cells which
CD3D80              CALL 803D,RIGHT          surround the given pixel.
CD1B80              CALL 801B,DOWN
CD1B80              CALL 801B,DOWN
CD3580              CALL 8035,LEFT
CD3580              CALL 8035,LEFT
CD0080              CALL 8000,UP
CD0080              CALL 8000,UP
78                  LD A,B               A:= number of cells surrounding pixel.
E1                  POP HL
C1                  POP BC               Restore HL and C to point to given pixel.
CBBC                RES 7,H              Point HL into screen (rather than copy).
FE02                CP 02
380C                JR C,NO_CELL         Jump if less than two cells surround pixel.
FE04                CP 04
3008                JR NC,NO_CELL        Jump if four or more cells surround pixel.
FE03                CP 03
2008                JR NZ,CONTINUE       Jump if exactly two cells surround pixel
79       CELL       LD A,C               A:= pixel position.
B6                  OR (HL)              Prepare to set pixel to create cell.
1803                JR PUT
79       NO_CELL    LD A,C               A:= pixel position.
2F                  CPL
A6                  AND (HL)             Prepare to reset pixel to remove cell.
77       PUT        LD (HL),A            Set or reset pixel as required.
CBFC     CONTINUE   SET 7,H              Point HL back into screen copy.
CB09                RRC C                Right one pixel (assuming within chr square).
30C5                JR NC,LOOP           Jump back if within character square.
CD4180              CALL 8041,RIGHT_1    Right one pixel (otherwise).
7D                  LD A,L               A:= low part of address.
E61F                AND 1F
20BD                JR NZ,LOOP           Jump back unless off right edge of screen.
CD1B80              CALL 801B,DOWN       Down one pixel.
7C                  LD A,H               A:= high part of address.
EEC0                XOR C0
B5                  OR L
20B4                JR NZ,LOOP           Jump back unless whole screen done.
C9                  RET
```

BASIC program to accompany machine code:

```
1000 RANDOMIZE USR 32845
1010 GO TO 1000
```

Suggested initial configurations:

Square:

```
10 PLOT 88,48
20 DRAW 80,0
30 DRAW 0,80
40 DRAW -80,0
50 DRAW 0,-80
```

Cross:

```
10 FOR i = USR "A" TO USR "A" + 7
20 POKE i,128
30 NEXT i
40 PRINT #0; AT 0,16; "graphic-A"; TAB 16; "graphic-A"
50 PLOT 0,80
60 DRAW 255,0
70 PLOT 128,0
80 DRAW 0,175
```

Circles:

```
10 FOR i = 10 TO 80 STEP 10
20 CIRCLE 128,80,i
30 NEXT i
```

Rays:

```
10 OVER 1
20 FOR i = -128 TO 127 STEP 3
30 PLOT 128,0
40 DRAW I,175
50 NEXT I
60 OVER 0
```

Strange symbols:

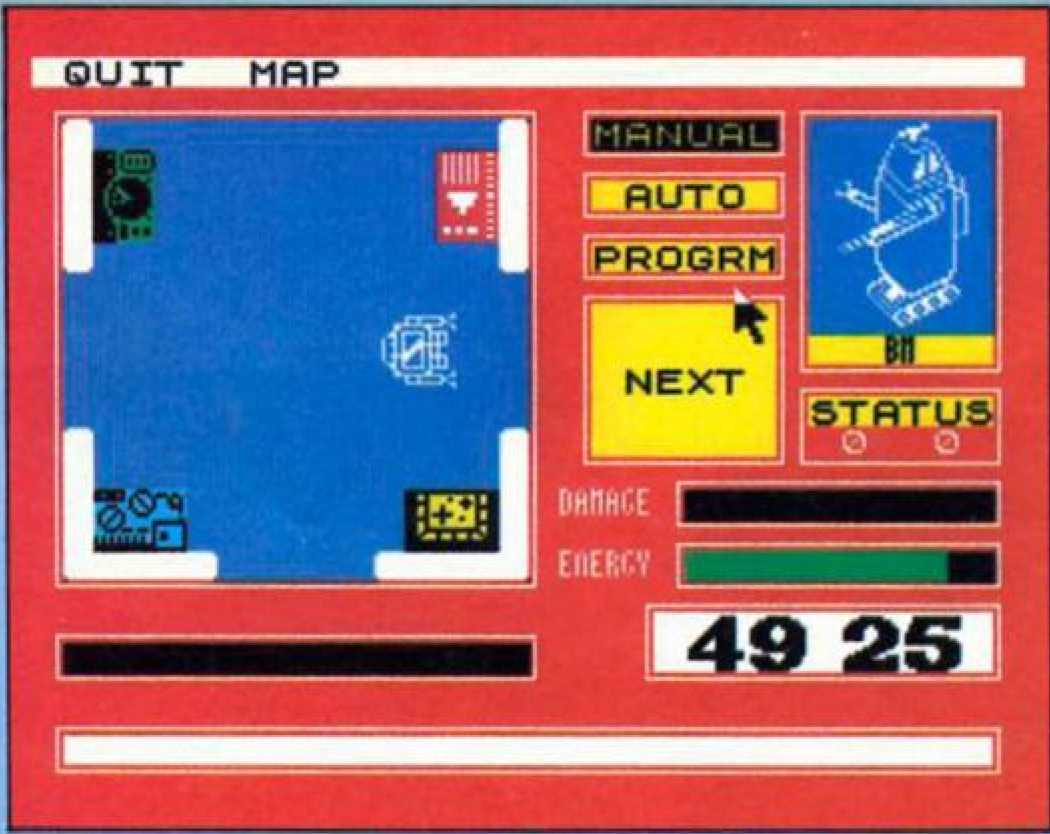
```
10 FOR i = 1 TO 4
20 LIST
30 NEXT i
```

Grid:

```
10 FOR i = USR "A" TO USR "A" + 6
20 POKE i,1
30 NEXT i
40 POKE i,255
50 PRINT #0; AT 0,0;
60 FOR i = 1 TO 704
70 PRINT "graphic-A";
80 IF i< 65 THEN PRINT #0;"graphic-A";
90 NEXT i
```

Frets:

```
10 FOR i = 60 TO 140 STEP 20
20 PLOT 0,i
30 DRAW 255,0
40 NEXT i
50 FOR i = 16 TO 256 STEP 32
60 PLOT i,60
70 DRAW 0,79
80 NEXT I
```



Time to save the world again as a planet sized U.F.O. is on collision course with earth.

Mission ΩMEGA

Mission Omega
Mind Games
£9.95

Everyone dreaded ever receiving a Priority 1 order from Star Fleet Command. It was the regular nightmare of a certain Captain Allen. Thankfully, though, Captain Allen didn't control a big fighter, or a space carrier, so he wouldn't get a Priority, would he?

So his mission had been a failure, even Star Fleet Command wouldn't execute a captain of a mining ship, would they? Surely not. Flying back from the outer reaches of space, all these thoughts, together with many more, were skipping about Captain Allen's mind.

How could he be expected to know that at that very minute, an unidentified flying object, roughly the size of earth's moon, was rushing at 99.9% the speed of light, towards his home, earth. He could also hardly have guessed that at a general council meeting, the Ambassador for the Ukraine had just passed a resolution stating that a small ship called The Windwraith should examine the object before it was blown apart by missiles.

One thing Captain Allen did know was that his ship was called The Windwraith, and he had just received a priority 1 order.

This is where you take control of the mission, now called Omega after the U.F.O. Having selected whether you wish to use joystick or keyboard, you are then ready to take on this challenge. You have one hour — in total — to make the Omega inactive, by shutting down all four reactors.

As if this were not enough, you must then escape, with all your crew. If you succeed, you will probably be heralded as a hero, given a Knighthood, two million pounds, etc. If you fail, well it could be worse, but not much.

Mission Omega is Argus' latest attempt to have a hit, and they may just have cracked it. Unlike many games, it cannot be fairly categorised, as it combines arcade adventure, with strategy and shooting. And all this is icon driven. Most of these are pretty obvious, with a bin halting any intended moves, and the rest being labelled.

At the top of the first screen are four icon headings, and to start with you must choose to build some robots, with which to map the Omega, and accomplish your increasingly difficult task. Once you have chosen to build a robot, you are given various different options. Firstly you must choose the robot you wish to use as a base unit. Some are fast, but can only carry a little weight. Consequently there are others that can carry anything, but are very slow.

Having chosen the robot, you must choose a weapon, a navigational device, and a power pack. Again, there are different choices in each category, with weight being the vital factor. However much you would like a fast robot with infra-red sensors, missiles and a nuclear battery, this is just not possible!

Having decided on the first robot, you must name him. If you wish to start playing, you may however, build more than one robot. As you find out when you first embark into the spaceship, having a few robots is a very necessary precaution.

Controlling the robots is a matter of choice, and position. The three options are automatic, program, or manual. Using automatic is good very early on, but to get anywhere it is necessary to either program a previously used route, or to do it all manually. Getting lost is a very real problem, so sticking together and using the map is essential for the first few minutes.

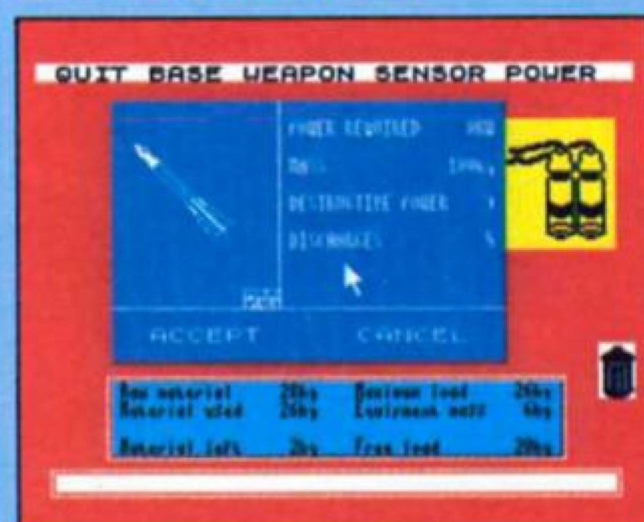
As the game is played, so the screen is updated depending on which robot you are currently controlling. At any time, you can swop from one robot to another, and indeed this is essential to allow each robot's batteries to recharge. Also displayed on screen is the ever decreasing amount of time you have to finish the mission, or die (mostly the latter).

Most of the U.F.O. is made up of separate rooms and corridors, with the odd computer, and even some baddies, who range from very easy to dispose of, to pretty impossible. Both the graphics and the sound are good, especially animation of the robots, which enables them to be distinguished at first glance.

Mission Omega is one of the best games to come from the Argus stable of companies, in this case Mind Games. It combines neat graphics with some excellent basic ideas, and comes up with a game that should appeal to everyone willing to spend more than ten minutes trying to save the world from imminent destruction.



GREAT



TRAILBLAZING

Sheffield based software house Gremlin are putting the finishing touches to four very different games. ZX blazed a trail northwards and found an outstanding arcade game.

Arthur Scargill is not known for his support of the software industry but indirectly he was responsible for giving Gremlin a helping hand when they were starting out in business back in 1984. Monty Mole, one of Gremlin's first games featured an effigy of Arthur and the media picked up on it with coverage of the game given on BBC and ITV news.

"We've really grown from there," said Gremlin director, Ian Stewart "and after Monty we concentrated on animal characters like Potty Pigeon, Sam Stoat and Monty sequels. From there we decided to diversify with games like Bounder, Way of The Tiger and Jack the Nipper."

Since 1984 Gremlin has expanded to the point where it now has 12 full time programmers working together in Sheffield and other smaller teams working elsewhere on outside projects." I would imagine that by next year we will have about 30 programmers on the payroll," said Ian.

Gremlin's immediate concern is gearing up for the Christmas market and they have four games lined up, an original arcade game, a combat adventure, a football strategy game and an arcade adventure.



Avenger: the loading screen



Trailblazer

Trailblazer

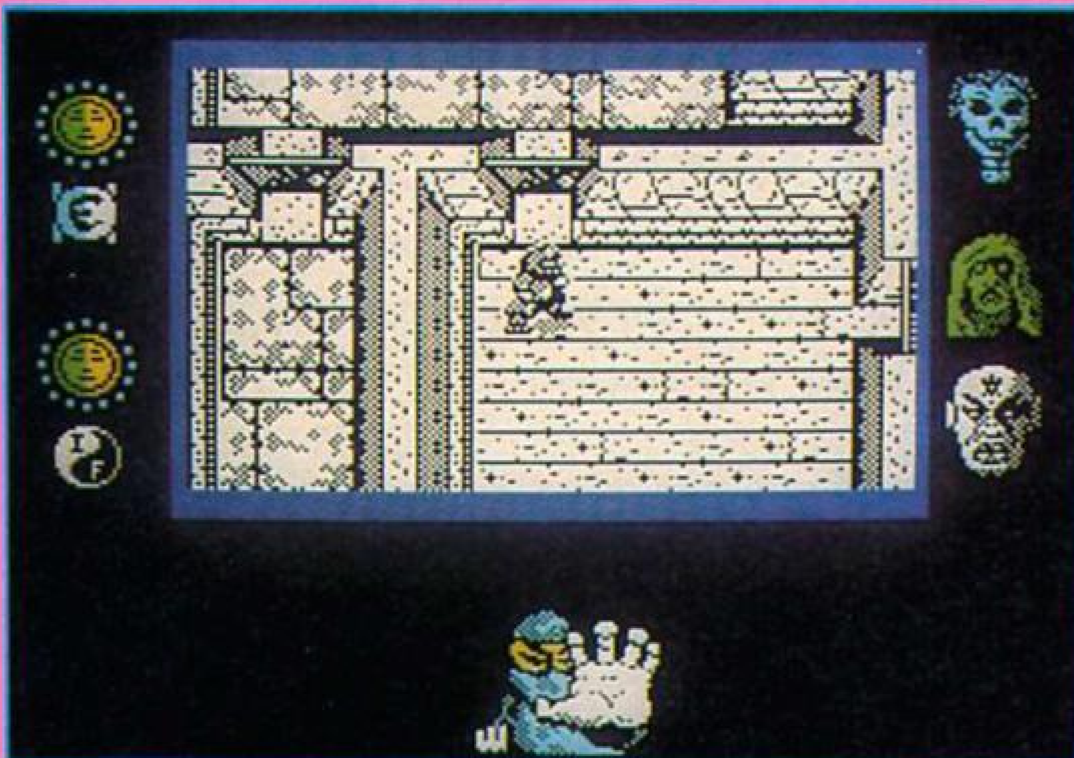
Of these, Trailblazer, a game that looks as if it is fresh from the coin-op arcade is now complete and it is immediately impressive from the first tug of the joystick. The idea behind the game is very simple, so simple you wonder why nobody had thought of it before. You control a ball and have to manouevre it down an ever changing path that rushes towards you at an alarming speed. Some squares slow you down, others give you the ability to bounce and the black spaces are conveniently placed for you to fall through once you think you've cracked the game.

There are 18 levels all with differing lengths and layouts which will test the reflexes and skill of even the most accomplished arcade fan. A nice feature is that you can practice any three levels before you have a go at the whole course and it's essential to familiarise yourself with as much as possible to give yourself a fighting chance. All runs are timed and you need to accomplish each level within a time limit in order to progress to the next level. If you make it within the time limit you get bonus time to carry over onto the new course.

Gremlin programmer, Greg Orme, who wrote Jack the Nipper showed me the finer points of trailblazing after the bouncing ball under my control had fallen off the path with embarrassing regularity. "The main thing," said Greg as he wrestled with the joystick, "is to build up a rhythm because if you miss a jump it's difficult to make up the time."

In keeping with one of the unbreakable laws of demonstrating games the bouncing ball disappeared

WITH GREMLIN



Avenger screen near to completion

through a hole in the road. "It's funny how that always happens when you are showing someone a game," said Gregg.

Trailblazer looks set to become an instant classic when it's released in November and although I'm wary of predictions I would say that unless the charts are abolished Trailblazer will be a number one.

Avenger

Avenger, the follow-up to Way of the Tiger is nearing completion. Having polished up on your Ninja skills in the three stages of Way of the Tiger you are now ready to start your mission of vengeance. Your foster father Naijishi, has been killed by Yaemon, Grandmaster of flame who has also stolen the Scrolls of Kettsuin from the temple. You have sworn to the God Kwon that you will avenge Naijishi's death and recover the scrolls.

There are six levels to explore and Gregg explained that in programming the game the emphasis had been on making the Ninja figure move around the locations quickly as most games of this sort tend to be very slow. "He's a Ninja after all so you can't have him hanging about. We've left out things like diagonal movement so he can get about quicker."

The graphics are impressive even at this stage in producing the game and a lot of incidental detail is included. The actual screen size of the animated acting is smaller than in similar games but it does not detract from the actual gameplay.

"The reason for having the smaller screen," said Greg, "is that you can get a larger map into the game and you have more memory to play around with for characters and graphics."

Without giving anything away about completing the game Greg said, "There's a lot of nasties to contend with and finding a monk is important but there's quite a few roaming around and you have to find the right one before you can go on.

There are three major villains in the game and each of their faces are set beside the main screen. When you kill one of them there is a nice, if ghoulish, touch as the face gradually sheds its skin to reveal the skull beneath. Although difficult to evaluate in its present form Avenger certainly looks as if it will be worth waiting for.

Gremlin have two other games in the pipeline which weren't available for preview — Footballer of the Year and FutureKnight. Although football strategy games would appear to be well trodden turf, Ian Stewart is confident that Footballer of the Year will be breaking new ground. "There's still a lot of mileage left in football based games and this game is different in that it's taken from the individual footballer's point of view. You are really running yourself as a business so that you can transfer yourself to Spain or Italy to get cash and prestige."

The aim of the game is to further your career from the lowly starting point of a 17-year-old Fourth Division player with £500 in cash to the pinnacle of success — the footballer of the year award.

FutureKnight, described by Gremlin as a "Blastventure" is a retrieve the Princess story ranging over various locations such as castles, spaceships and forests. Spegbott the terrible has the princess Amelia in his clutches and you have to teleport aboard a spacecraft, the SS Rustbucket where she was last seen. There are twenty levels to battle through against Berserka robots before finally reaching Castle Gremlin on a nearby planet where a final showdown with the Henchodroid protecting the princess awaits you.

A decision was being made that day on whether the Spectrum version of FutureKnight would include a scrolling screen. "At the moment we haven't finally decided whether it will scroll or use a flick screen between locations. In the end the decision will be made on whatever's best for the product. Although scrolling would be aesthetically pleasing the gameplay may be improved if we use flick screen. "We'll see," said Ian, "mind you it would be a shame if we lost the scrolling."

Help!

Gremlin have found that the games business doesn't end with just producing games. An "after sales" service has developed for harrassed games players stuck in the middle of Gremlin games. "We've experienced this with games before but nothing like the response we've had with Jack the Nipper. There's always a danger that games will be played for a day and if they get nowhere they won't pick it up again. Giving people hints if they are stuck increases their enjoyment of the product."

"It's always difficult pitching the degree of difficulty of a game. I think that in some of our past games not enough information has been given and they have been too hard. On the other hand you don't want to make it too easy.

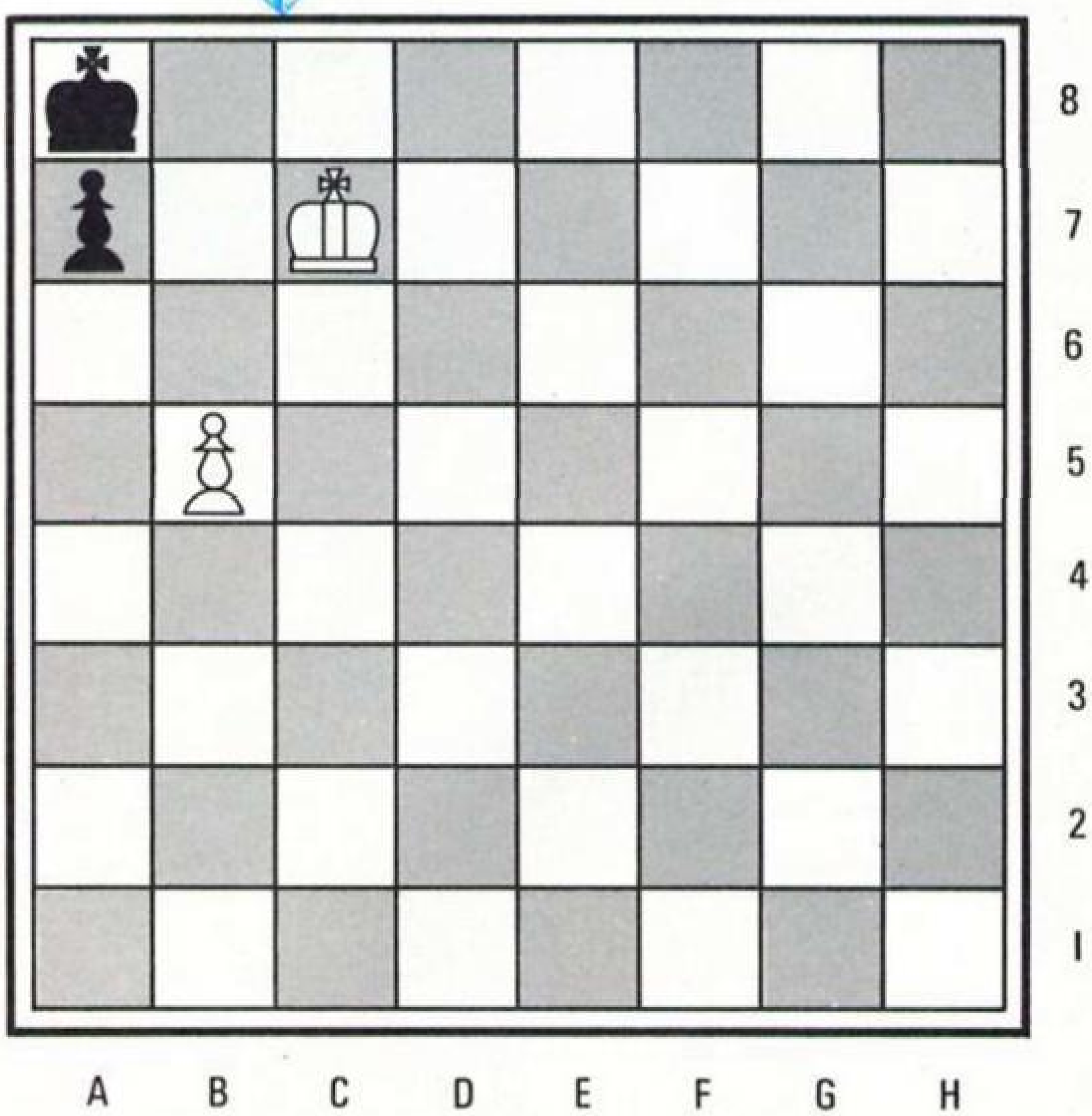
With Trailblazer, at last, Gremlin seem to have solved the dilemma. No one will complain that it's too hard-they will be too busy playing it.

CAPTURE COLOSSUS

Here's your chance to capture Colossus Chess 4. There are 20 copies of this superlative program from CDS just waiting to be won — but there is a problem...

COMPETITION





Colossus Chess 4 is a formidable/invincible chess adversary depending on your level of play and the level at which you choose to play it. We've got together with CDS to give you the opportunity of carrying off Colossus and all you have to do is solve a chess problem set by the writer of Colossus, Martin Bryant.

The Problem

Look at the endgame situation on the board illustration. You are playing the white pieces. White to move, and checkmate black in six moves. What you have to do is note down white's first move from this situation to ensure victory in six moves. Please write your answer using the board coordinates, for example Pawn B5-B6. Write your answer on the coupon provided and also clearly on the back of your entry envelope. It's your move...

The competition is open to all ZX readers except employees of Argus Specialist Publications, Alabaster Pasmore and CDS. Post your entries to Colossus Competition, ZX Computing Monthly, No 1 Golden Squaqre, London W1 3AB. The closing date for entries is November 7th 1986.

COLOSSUS COMPETITION

White's next move is ....

Name ....

Address ....

....

Post this coupon to Colossus Competition, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB.

CROSSWIRES

Ray Elder presents more readers' problems — and solutions.

Brother bother

Dear Sir,

Q *I own a Spectrum+ with a Brother EP44 printer. The interface I use is the ZX Lprint 3 RS232 interface.*

At the moment I am having difficulties configuring Tasword 2 and Masterfile to run with the interface.

Could you please provide me with precise instructions on how to incorporate the Lprint 3's commands into both these programs.

Your assistance would be greatly appreciated. Thank you.
Cheam Kong Whee,
Kuala Lumpa,
Malaysia.

A A very nice set up indeed and it should operate without any real problems. I assume that you have ZX Lprint's own RS232 lead or that you are certain that the one you have is wired correctly, so let's look at Tasword 2 first and take it in easy steps.

1. LOAD Tasword 2.
2. Use the "STOP" option to get to the menu.
3. Select 'B' — Go into Basic.
4. Type LIST 15 and press ENTER.
5. Press key N in reply to the Scroll? prompt.
6. Hold Caps Shift and press key 1 to EDIT. The line will appear at the bottom of the screen.
7. Add to the start of the line, after the line number, LPRINT ; CHR$ 5; CHR$ 0; "S";

(Note: the rest of the line must follow on from here and the ' character is essential). Now press Enter to insert the line in the program.

8. Type GO TO 25.
9. Now select menu option 'G'.
10. Press ENTER to move out of the define graphics page.
11. Answer 'Y' to change control codes.
12. Enter 0,0,0,57855, one at a time, for the four codes and leave the rest by just pressing ENTER for each question.
13. Use option 'T' and SAVE your modified program for future usage.

OK? So now on to Masterfile.

Masterfile cannot be modified so easily. First you will have to determine which of the sequences CHR$ 0; CHR$ 1,2,3 or 4 operates best with your printer in graphics mode. I use CHR$ 0; CHR$ 4. Once you have done this then load MF and use one of the SAVE options. Then BREAK the program before or during the actual save. (There is no need to use a recorder).

Change or modify line 4010 to read: 4010 LPRINT: LPRINT CHR$ 0; CHR$ 4; CHR$ 0; "S"; CHR$ 13: COPY: GO TO USR R.

This will allow a simple page image copy, to tailor the program to take advantage of the full 80 columns and various layouts will involve using more advanced programming and the drops in and out of basic to line numbers, a sort of interrupt system, is explained in the manual. It will also depend on the layout and contents of each file and would take at least a page in itself to explain. I am afraid that you will have to study the manual and experiment.

Finally type GO TO USR R and then SAVE a copy of your modified program (option V).

A strange one

Dear Sir,

Q *I would be grateful for any advice that you can offer on a problem that I have with my computer equipment.*

I use a Spectrum+ with an Opus Discovery 1. After a few weeks use the STOP keyword developed an intermittent fault, and subsequently failed altogether. Since this keyword is used in Tasword 2 and Vu-File, this fault is very annoying.

The keyboard is fine when the Spectrum is not connected to the Discovery and, just to add to the confusion, works OK with a ribbon cable attached to the expansion slot, but not when the multiface or joystick is attached to the cable.

The Discovery works OK with another Spectrum, so I'm fairly sure that isn't the problem.

I've sent the Spectrum to Mancomp for repair, but they merely replaced the keyboard membrane and ignored my comments about faults when connected to peripherals. The fault persists!
Yours faithfully,
W. J. Thomas,
Newtown, Powys.

A I have a friend who is an electronics engineer and I took this letter to him, his reply was unprintable. Apparently this is a problem which is rather more frequent than is realised, and it is very time consuming (and therefore expensive) to check out. He showed me a BBC that worked perfectly, yet showed up on his test equipment as faulty.

It seems that on occasion two or more units that in all ways should be compatible just will not operate together. It could be a small hairline crack on a PCB which only fails when it is connected to a particular piece of equipment because that is slightly out of alignment and it puts pressure on it, or a chip or diode which only fails after a minute drop in the power level occurs, usually due to a peripheral, as in using the Opus built in PSU.

Yes there **is** always something at fault, but tracking down the precise reason is a lengthy and frustrating job. If you must have it done, and I would be most unhappy to let such a situation continue, then your complete system will have to be checked as it could be any one or more of the units which is causing the failure.

Take the lot to a reputable engineer and be prepared for a long wait and a big bill. It could exceed the cost of a new computer! You are one of a number of rather unlucky owners and you have my sympathies.

128 Printer problems

Dear Sir,

Q *Would you please help me operate my printer. I, like one of your readers, bought a Spectrum 128 with a serial 8056 Compact printer. My problem is with the printer codes. I can get a print out in normal text mode with FORMAT "P" 1200; this however does not work in the 48K mode.*

So could you please explain how to operate the codes and how to operate the printer when in 48K mode, the small booklet you get with the printer shows the codes but not how to use them. I'll be very grateful if you could help me.
I. S. White, Tonbridge.

A Meanwhile a mysterious reader who didn't include his/her name and address sent us a part solution. ZXC has the most helpful and skilful readers and we really appreciate your help and involvement!

I hand over to M. Mysterious: In Crosswires (July), two people had problems with the Spectrum 128 RS232 port.

First you need a proper Spectrum lead, I have a Brother M-1009+128 Spectrum and I got the lead from RSD Connections Ltd. Dept ZXC, PO BOX 1, WARE, HERTS. It costs £12.08 and they have a wide selection of accessories.

To operate the printer so that we can send control codes we have to resort to a short machine code program. This LOADs the A register with the code to be sent and then calls some built in routines to get it to the printer. I give a small program to show how to use it.

```
10 LET P$=CHR$ 27+CHR$ 52+"Testing Italics Mode"
20 GO SUB 9000
30 STOP: REM or rest of program
9000 FOR a=1 TO LEN P$
9005 POKE add+1,CODE P$(a)
9010 RANDOMIZE USR add
9020 NEXT a
9030 POKE add+1,10: RANDOMIZE USR add
9040 POKE add+1,13: RANDOMIZE USR add
9050 RETURN
9990 LET add=65368
9991 FOR a=add TO add+10: READ n: POKE a,n: NEXT a
9992 DATA 62,0,205,0,91,205,39,1,195,0,91
9993 GO TO 10
```

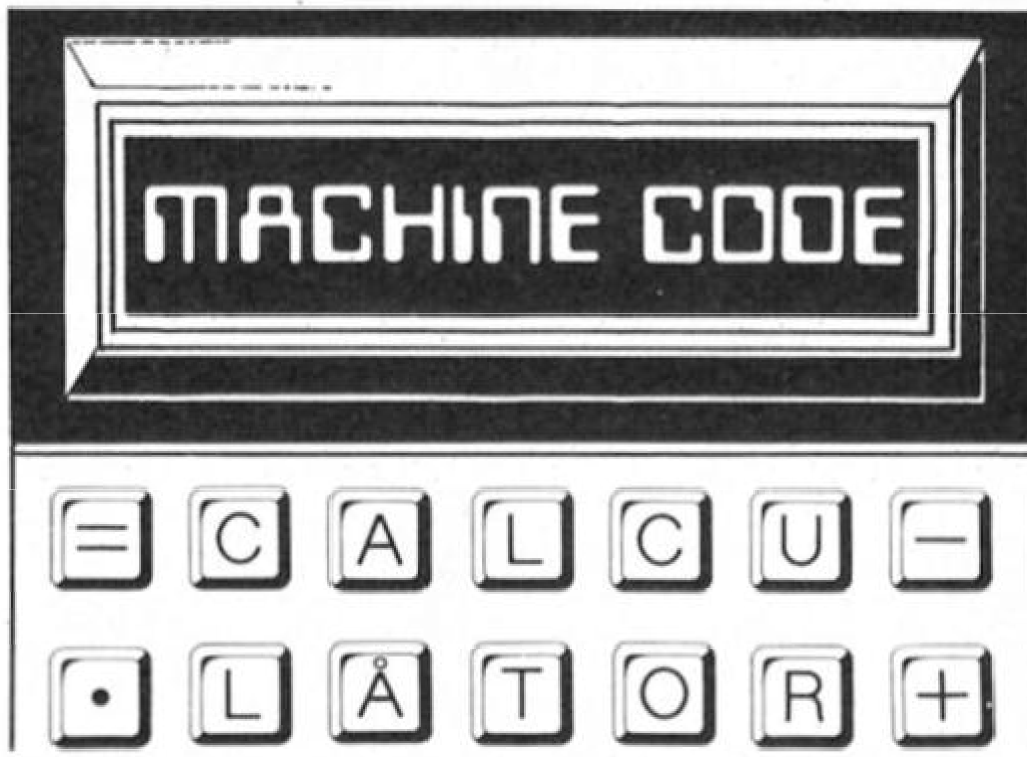
M/c Program

The code is located in the UDG area but can be put anywhere you wish, at the moment you can only use it in 128 mode and once you have typed it in start by going to line 9990 (GO TO 9990).

In line 10 you should set up P$ to contain the codes you want to send. Once the printer mode has been set then you can print in the usual way until you want to change the mode. Then you have to set up an appropriate P$ and GO SUB 9000 again.

Lines 9030 and 9040 do a carriage return and line feed, you may wish to omit them.

I use the variable add which is set to the address at which I store the code, and to send a code or character you have to POKE its CODE into address add+1.



More explorations of the Spectrum's calculator functions in Part 4 of Toni Baker's series.

There are many questions still unanswered. How do you get random numbers (the RND function)? How can you calculate SCREEN$(X,Y)? How do you read the value of a BASIC variable? How do you slice a string? These questions, and many more, will now be answered.

POINT: The machine code instruction CALL POINT_SUB (address 22CB) will pop two numbers, X,Y, from the calculator stack, evaluate POINT(X,Y), and will put the result back onto the calculator stack.

ATTR: The machine code subroutine S_ATTR_S at address 2307 will take two numbers, Y,X, from the calculator stack, and replace them with ATTR(Y,X).

SCREEN$: The machine code subroutine S_SCRN$_S at address 2535 will take two numbers, Y,X, from the calculator stack, and replace them with the string SCREEN$(Y,X). The SCREEN$ bug which is present in BASIC is fortunately *not* present in this machine code routine, so you can happily use it with no problems (In BASIC, the expression "★" + SCREEN$(0,0) will almost invariably give the wrong answer — try it by printing varous different things as position 0,0!).

The SCREEN$ function will only detect ASCII characters. It will not detect either block graphics or user defined graphics. By altering the value of the system variable (CHARS) to (UDG) — 100h prior to calling S_SCRN$_S it is possible to search for the user defined graphics instead of the ASCII characters. You must restore (CHARS) after the subroutine call. The value returned by the subroutine will have character code 70h less than the UDG found.

More comprehensively, the program which was listed in last month's ZX Computing in the article "Elementary Graphics Part Three" will detect and correctly return all characters.

SLICING: Figure One lists a program which I have called SLICE. Its purpose is to slice strings, in the same way that you can in BASIC. To slice a string simply CALL SLICE (which must of course be written into the Spectrum's memory at some convenient location) and the job will be done.

The subroutine will remove three items from the calculator stack, A$,X,Y, and will replace them with the sliced string A$(X TO Y). Also in Figure One is a program called SLICE_INT which only requires one item (the string to be sliced), A$ say, on the calculator stack, and will replace it with the sliced string A$(BC TO DE).

RND: The routine labelled RND in Figure Two will evaluate the RND function, leaving a random number between 0 and 1 as a new entry at the top of the calculator stack. There is no real need to write the routine as a subroutine, since it is so short you might just as well include it as it stands in your programs.

As with the BASIC RND function it suffers from the disadvantage that it is incredibly slow. For this reason Figure Two contains two other programs, labelled RANDOM and RNDQ respectively. The purpose of RANDOM is to assign (SEED) with the next value in the pseudo-random sequence. On return HL will contain the new value of (SEED), being a random word between 0000 and FFFF; also, the A register will contain the high part of this number, and will therefore be a random byte between 00 and FF.

The RNDQ subroutine (which relies on the existence of the RANDOM subroutine) will do *exactly* the same job as RND — ie will leave a random number between 0 and 1 at the top of the calculator stack — but will do it much, much faster.

PI: Since the calculator has a built in procedure for calculating PI/2, the calculation of PI is particularly easy. Figure Three contains four different routines for calculating PI — use whichever of the three you want. They are listed in order of speed — the first is the slowest.

INKEY$: INKEY$ (as opposed to INKEY$ #N) is particularly easy to do in machine code. The program listed in Figure Four will do the job nicely, placing the string result of an immediate keyboard scan at the top of the calculator stack. INKEY$ #N, of course, needs no special routine to do that.

Figure 1

```
CD991E   SLICE       CALL 1E99,FIND_INT2     BC:= second slice parameter.
C5                   PUSH BC                 Store on machine stack.
CD991E               CALL 1E99,FIND_INT2     BC:= first slice parameter.
C5                   PUSH BC                 Temporarily store on stack.
CDF12B               CALL 2BF1,FP_TO_AEDCB   DE:= address of string to slice.
AF                   XOR A                   A:= zero.
EB                   EX DE,HL                HL:= address of string.
D1                   POP DE                  DE:= first slice parameter.
C3962A               JP 2A96,SL_DEFINE_2     Jump into ROM routine to
                                             complete the task.

D5       SLICE_INT   PUSH DE                 Stack second parameter.
C5                   PUSH BC                 Temporarily stack first parameter.
CDF12B               CALL 2BF1,FP_TO_AEDCB   DE:= address of string.
AF                   XOR A                   A:= 0.
EB                   EX DE,HL                HL:= address of string.
D1                   POP DE                  DE:= first parameter.
C3962A               JP 2A96,SL_DEFINE_2     Slice the string.
```

Communicating with BASIC

Using machine code throughout is good practice, but sometimes it makes things easier if you can use numbers and strings on the calculator which were specified in BASIC. Better still if you could use machine code results in BASIC. There are a number of ways to do this so let's look at the easy ways first.

Consider the following BASIC statement: LET P=Q+USR address

This will call a machine code subroutine at the specified address, but with a rather interesting property. At the time the machine code is executed the parameter Q will be at the top of the calculator stack. This is really the easiest way of passing a numeric parameter from BASIC to machine code. I'll give you an example — a little routine which works out SQR (Q+1). In other words, you will be able to use LET P=Q+USR address in place of LET P=SQR(Q+1). You will notice that this technique firstly accepts a numeric parameter passed from BASIC, and finally passes a numeric parameter calculated in machine code back to BASIC.

The example I mentioned is listed in Figure Five. The calculation of SQR(Q+1) is really quite simple — since it is the parameter passing which is the point of the example. There are some things to watch out for:
LET P=Q+R+USR address will pass Q+R to machine code
LET P=(expression)+USR address will pass the value of the expression to machine code
LET P=USR address+Q will not work!

There are many other ways of talking of machine code from BASIC, and vice versa. I'll illustrate one such method with a routine which returns the difference in lengths of two strings. The all important BASIC instruction is this:
DEF FN A(P$,Q$)=USR address.

The trick is to write a machine code routine at the given address in such a way that whenever FN A(string expression, string expression) is encountered it will be regarded as ABS(LEN string−LEN string).

How we do this is to look at the way that BASIC treats the arguments of its user defined (FN) functions. It stores in memory a list of records, with one record for each parameter required by FN, and in the same order, with eight bytes for each numeric parameter and nine bytes for each string parameter. Figure Six illustrates the format for both types of record, while Figure Seven lists two programs relevant to this technique. The first program performs the task I have mentioned as an example (ie it calculates ABS (LEN

Figure 2

```
3D                       DEC A                 Adjust for special case.
5F          RAND_MOD     LD E,A
1600                     LD D,00
A7                       AND A
ED52                     SBC HL,DE             HL:= remainder modulo 65537
                                                    of 75*(SEED+1).
2B                       DEC HL                HL:= (75*(SEED+1) MOD 65537)-1.
7C          RAND_EXIT    LD A,H                A:= random byte.
22765C                   LD (SEED),HL          Store new value for system variable.
C9                       RET

CD????      RNDQ         CALL RANDOM           Find new value for (SEED).
44                       LD B,H
4D                       LD C,L                BC:= random word.
CD2B2D                   CALL 2D2B,STACK_BC    Place on calculator stack.
EF                       RST 28                Switch calculator on.
3D                        restack              Re-stack in floating point form.
38                        end calc             Switch calculator off.
7E                       LD A,(HL)             A:= exponent byte.
A7                       AND A
C8                       RET Z                 Return if random number is zero.
D610                     SUB 10
77                       LD (HL),A             Reduce exponent by sixteen, thus
C9                       RET                        dividing number by 65536.
061D        RND          LD B,1D                 B:= calculator code for VAL.
EF                       RST 28                  Switch the calculator on.
34B03F0201                stk data "RND"         Stack the string "RND".
3B                        execute B              Evaluate the string using VAL.
38                        end calc               Switch the calculator off.
2A765C      RANDOM       LD HL,(SEED)            HL:= previous random number seed.
7C                       LD A,H
A5                       AND L
3C                       INC A
2005                     JR NZ,RANDOM_2          Jump unless seed = FFFF.
21B5FF                   LD HL,FFB5              In which case new value is FFB5.
182A                     JR RAND_EXIT
AF          RANDOM_2     XOR A                   AHL:= SEED.
23                       INC HL                  AHL:= SEED+1.
54                       LD D,H
5D                       LD E,L                  DE:= SEED + 1.
29                       ADD HL,HL
8F                       ADC A,A                 AHL: = 2 * (SEED+1)
29                       ADD HL,HL
8F                       ADC A,A                 AHL:= 4 * (SEED+1)
29                       ADD HL,HL
8F                       ADC A,A                 AHL:= 8 * (SEED+1)
19                       ADD HL,DE
CE00                     ADC A,00                AHL:= 9 * (SEED+1)
29                       ADD HL,HL
8F                       ADC A,A                 AHL:= 18 * (SEED+1)
29                       ADD HL,HL
8F                       ADC A,A                 AHL:= 36 * (SEED+1)
19                       ADD HL,DE
CE00                     ADC A,00                AHL:= 37 * (SEED+1)
29                       ADD HL,HL
8F                       ADC A,A                 AHL:= 74 * (SEED+1)
19                       ADD HL,DE
CE00                     ADC A,00                AHL:= 75 * (SEED+1)
24                       INC H
25                       DEC H
2006                     JR NZ,RAND_MOD          Jump unless HL <0100.
BD                       CP L
2803                     JR Z,RAND_MOD           Jump if A = HL.
3801                     JR C,RAND_MOD           Jump if A < HL.
```

Figure 3

```
061D        PI_VERSN_1    LD B,1D              B:= calculator code for VAL.
EF                        RST 28               Switch calculator on.
34B0380201                stk data "PI"        Stack the string "PI".
3B                        execute B            Evaluate the string using VAL.
38                        end calc             Switch calculator off.

EF          PI_VERSN_2    RST 28               Switch calculator on.
A3                        const PI/2           Stack half of PI.
31                        duplicate
0F                        add                  Double it.
38                        end calc             Switch calculator off.

EF          PI_VERSN_3    RST 28               Switch calculator on.
A3                        const PI/2           Stack half of PI.
38                        end calc             Switch calculator off.
34                        INC (HL)             Increment exponent byte, thus
                                                 doubling the number.

EF          PI_VERSN_4    RST 28               Switch calculator on.
34F2490FDAA2              stk data 3·1415927   Stack constant PI directly.
38                        end calc             Switch calculator off.
```

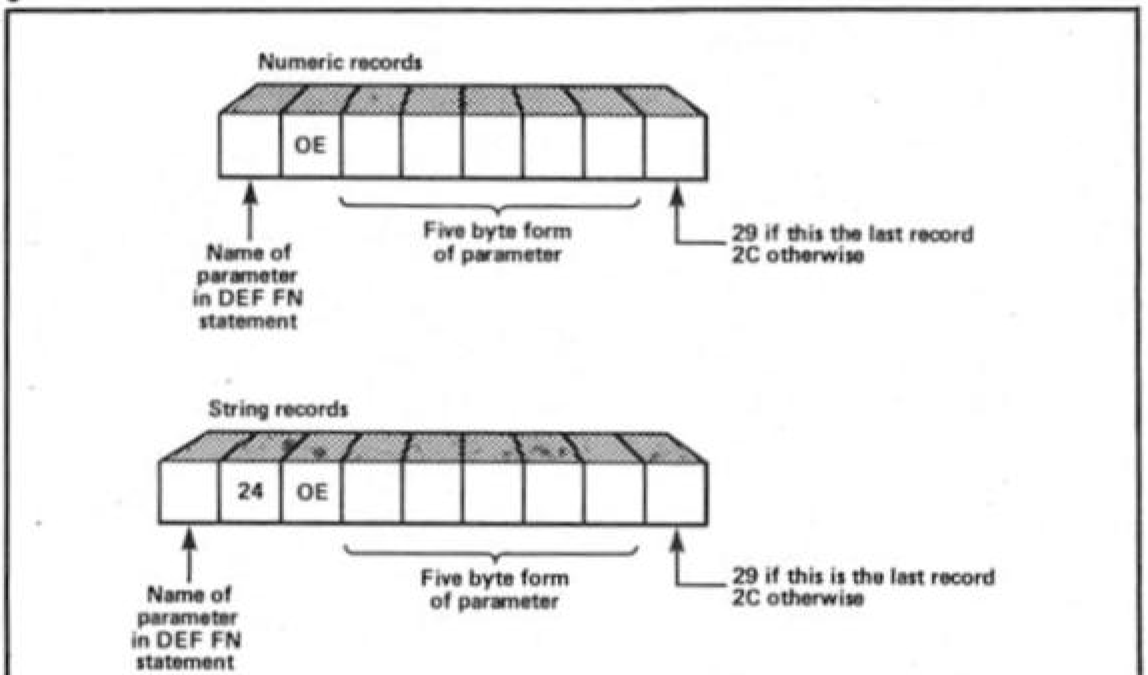
Figure 4

```
0618        INKEY$        LD B,18              B:= calculator code for VAL$.
EF                        RST 28               Switch calculator on.
34B0390201                stk data "INKEY$"    Stack the string "INKEY$"
3B                        execute B            Evaluate the string using VAL$.
38                        end calc             Switch calculator off.
```

5

```
EF          DEMO_1        RST 28               Switch calculator on. Note that
                                                 Q is already on the stack.
A1                        const one            Q,1
0F                        add                  Q+1
28                        sqr                  SQR (Q+1)
38                        end calc             Switch calculator off.
010000                    LD BC,0000           Return ZERO to BASIC via BC.
C9                        RET
```

6



string–LEN string).

The second program is more general — it will place on the calculator stack the value of each function argument in the proper code. It is intended to be used as a subroutine to be called by your own machine code programs addressed by DEF FN.

By using a combination of both of these techniques it is possible to construct a user defined function of more than one argument, and which returns a string result.

We have not looked at the possibility of returning string results as yet, but the technique is dead easy. It is similar to the LET P=Q+USR address technique, but here we have to use LET P$=Q$ AND USR address. In this case the machine code subroutine will be called with one item, the string P$, on the calculator stack.

If we alter the topmost item on the stack to a string of our own choosing, and return to BASIC with BC containing a non-zero value (so that AND will not empty our string) then P$ in the above example may be assigned with a string defined in machine code!

It follows then, that you should be able to combine these two techniques, and come up with a BASIC statement similar to this: **DEF FN A$(P$,Q)=P$ AND USR address.**

Of course you can have any number of arguments in the brackets — including none at all. The arguments can be either string or numeric. The part of the statement in between "=" and "AND" doesn't have to be P$ — it can be any string expression — even a string constant. On entry to your machine code routine the value of this string expression will be at the top of the calculator stack, whilst the arguments of the user defined function will be stored as usual as records pointed to by (DEFADD).

Suppose you wanted a user defined function which had two numeric arguments and returned a string result — you could then use DEF FN A$(P,Q)="" AND address. Your own machine code routine would replace the original (empty) string on the stack with a string of your own choosing.

Figure Eight lists a user defined function which returns a string result. The function is referred to as FN F$, and it is the Spectrum equivalent to the QL's FILL$ function. It requires two arguments — a string and a number, and it returns the given number of copies of the string, all joined together (eg FN F$("*",4) would equal "****").

And now for a numeric function. Once again this is a function which you cannot define in a BASIC DEF FN statement using BASIC alone. The function is called the Factorial function. It's a very easy function to define. The factorial of zero is one; the factorial of one is one; the factorial of two is 1 * 2 (=2); the factorial of three is 1 * 2 * 3 (=6); the factorial of seven is 1 * 2 * 3 * 4 * 5 * 6 * 7 (=5040), and so on. In general, the factorial of N (a positive integer) is the product of all the integers from one to N all multiplied together, or written down mathematically it is

1 * 2 * 3 * . . . * N. In mathematics the factorial of N is usually N! (ie N followed by an exclamation mark) and is pronounced N factorial. In BASIC, however, we are going to write it as FN F(N), and we'll define it in machine code using the calculator.

To define it you'll need the BASIC statement DEF FN F(N)=USR factorial somewhere in your BASIC program, and the program of Figure Nine at the relevant address.

Note that the program has a choice of two alternative endings. Version one is what we would expect; this is the normal way of ending things, with all the stacks balanced. This, however, has one big disadvantage — it can only return Integers (which can fit into BC). This means that we can only calculate up to eight factorial. The second version of the ending does not suffer from this disadvantage, and any numeric value whatsoever may be returned to BASIC.

The rules for using this, the best and simplest trick in all calculatordom, are as follows. The calculator stack must be unbalanced, with one item too many on the stack. It is the extra (topmost) item which will be returned to BASIC, so any number (not just integers) may be returned). The machine stack must also be unbalanced, with one item too few on the stack. The normal return address from USR will therefore be bypassed — this bypasses the subroutine STACK_BC, and hence causes the trick to work. The registers HL and DE must be correctly assigned, with DE=(STKEND), and HL=(STKEND)−5.
The "end calc" instruction which is used to exit from calculator code will leave HL and DE correctly assigned in this way. The shortest way to assign DE and HL in this manner if they are not already so set up is to use the sequence of instructions RST 28/"end calc" (in hex EF followed by 38).

With the second ending in place bigger numbers may be returned. You may also use this same technique to return decimals (non-integers). Now you have a complete factorial function defined as FN F. Perhaps the advantages of the calculator are now beginning to sink in....

Getting hold of BASIC variables

Getting BASIC variables onto the calculator stack is one of the most simple things imaginable. The first thing you have to do is to place the name of the variable, in the form of a string, at the top of the calculator stack. For instance, the string "A" for the numeric variable A, the string "B$" for the string variable B$, the string "TOTAL" for the numeric variable TOTAL, and so on. Then you must load the B register with either 1D (if the variable is numeric) or 18h (if the variable is string) and use the sequence of instructions RST 28/"execute B"/"end calc" (in hex EF 3B 38).

This works because 1D is the code for VAL, and 18 is the code for VAL$. In other words, you are effectively calculating VAL("A"), VAL$("B$"), VAL("TOTAL"), et cetera.

7

```
2A0B5C    LEN_DIFF   LD HL,(DEFADD)          HL: points to first record.
010600               LD DE,0006
19                   ADD HL,DE               HL: points to length of first str.
4E                   LD C,(HL)
23                   INC HL
46                   LD B,(HL)               BC:= length of first string.
1E08                 LD E,08
19                   ADD HL,DE               HL: points to length of 2nd string.
5E                   LD E,(HL)
23                   INC HL
56                   LD D,(HL)               DE:= length of second string.
EB                   EX DE,HL                HL:= length of second string.
A7                   AND A
ED42                 SBC HL,BC               HL:= difference in lengths.
44                   LD B,H
4D                   LD C,L                  BC:= difference in lengths.
D0                   RET NC                  Return if result positive.
78                   LD A,B
2F                   CPL
47                   LD B,A
79                   LD A,C
2F                   CPL
4F                   LD C,A
03                   INC BC                  Otherwise negate BC.
C9                   RET

2A0B5C    GENERAL    LD HL,(DEFADD)          HL: points to first record.
23        GEN_LOOP   INC HL                  HL: points to byte 0E or 24.
7E                   LD A,(HL)
FE0E                 CP 0E
2801                 JR Z,GEN_SKIP           Jump if record is numeric.
23                   INC HL                  Skip over extra byte.
23        GEN_SKIP   INC HL                  HL: points to five byte form.
CDB433               CALL 33B4,STACK_NUM     Push parameter onto calc stack.
7E                   LD A,(HL)
FE2C                 CP 2C
28F0                 JR Z,GEN_LOOP           Jump back except for last record.
C9                   RET
```

8

```
DEF FN F$(X$,X) = X$ AND USR fill
2A0B5C    FILL       LD HL,(DEFADD)          HL: points to first record.
110B00               LD DE,000B
19                   ADD HL,DE               HL: points to 5 byte form of number.
CDB433               CALL 33B4,STACK_NUM     Stack number of repeats.
CD991E               CALL 1E99,FIND_INT2     BC:= number of repeats.
C5                   PUSH BC
CDF12B               CALL 2BF1,FP_TO_AEDCB   BC:= length of str; DE:= address.
E1                   POP HL                  HL:= number of repeats.
E5                   PUSH HL                 Stack number of repeats.
D5                   PUSH DE                 Stack address of old string.
ED43AE5C             LD (5CAE),BC            Store length of old str in Mem 5.
50                   LD D,B
59                   LD E,C                  DE:= length of old string.
CDF82A               CALL 2AF8,GET_HL*DE_1   HL:= length of new string.
44                   LD B,H
4D                   LD C,L                  BC:= length of new string.
```

▶ ▶ ▶

SPECTRUM PROGRAMMING

8

```
78                          LD A,B
B1                          OR C
2005                        JR NZ,NON_EMPTY            Jump unless new string is empty.
F1                          POP AF                     Delete address of old string.
F1                          POP AF                     Delete number of repeats.
C3B62A                      JP 2AB6,STK_AEDCB          Stack empty string and exit.
F7          NON_EMPTY       RST 30                     DE: points to space for new string.
CDB62A                      CALL 2AB6,STK_AEDCB        Stack new string parameters.
E1                          POP HL                     HL:= address of old string.
C1                          POP BC                     BC:= number of repeats.
C5          FILL_LOOP       PUSH BC
ED4BAE5C                    LD BC,(5CAE)               BC:= length of old string.
E5                          PUSH HL
EDB0                        LDIR                       Make next copy of old string.
E1                          POP HL
C1                          POP BC
0B                          DEC BC                     BC:= number of remaining repeats.
78                          LD A,B
B1                          OR C
20F1                        JR NZ,FILL_LOOP            Loop back for all copies.
03                          INC BC                     Return to BASIC the value one.
C9                          RET
```

9

```
DEF FN F(N) = USR factorial

2A0B5C      FACTORIAL       LD HL,(DEFADD)             HL: points to argument record.
23                          INC HL
23                          INC HL
CDB433                      CALL 33B4,STACK_NUM        Push argument onto calc stack.
CD941E                      CALL 1E94,FIND_INT1        A:= this argument.
A7                          AND A
2001                        JR NZ,FCT_POSITIVE         Jump unless N equals zero.
3C                          INC A                      Set A to one, since 0! = 1!
CD282D      FCT_POSITIVE    CALL 2D28,STACK_A          Push back onto calc stack.
EF                          RST 28                     Switch calculator on.
31                           duplicate                 The calc stack now contains two
                                                       identical numbers, each equal to
                                                       N. If we LET K = N, and if we
                                                       take M*...*N to mean the product
                                                       of all integers between M and N
                                                       inclusive, then the items on the
                                                       calc stack may be written:

                                                       K*...*N,K
A1          FCT_LOOP         const one                 K*...*N,K,1
03                           subtract                  K*...*N,K-1
31                           duplicate                 K*...*N,K-1,K-1
30                           eq zero                   K*...*N,K-1,K=1?
0006                         jump true,FCT_EXIT        Jump if K equals one.
C5                           store M5                  K*...*N,K-1   (M5 contains K-1).
04                           mult                      (K-1)*...*N
E5                           recall M5                 (K-1)*...*N,K-1
33F6                         jump FCT_LOOP             LET K = K - 1 and loop back.
02          FCT_EXIT         delete                    1*...*N
38                           end calc                  Switch calculator off.
VERSION ONE ENDING:
CDA22D                      CALL 2DA2,FP_TO_BC         BC:= required factorial.
DAAD31                      JP C,31AD,REPORT_6         Error 6 if too big for BC.
C9                          RET
VERSION TWO ENDING:
C1                          POP BC                     Bypass STACK_BC subroutine.
C9                          RET
```

Note that you should not use "val" or "val$" directly as calculator instructions, unless B is also assigned with 1D or 18 as before — this is because the calculator instructions "val" and "val$" are B register dependent (see Appendix in last month's article).

With single letter variable names, the process is even easier, because the technique used to create the variable name string is so easy. All you have to do is place an integer on the calculator stack — the character code of the letter of the variable name — and use the calculator "chr$" function to turn it into a string. Figure Ten contains two programs which demonstrate this idea — the first puts the value of the BASIC variable A onto the stack, and the second, by a similar technique, puts the string value of the BASIC variable B$ onto the stack.

Figure Eleven consists of a table, giving you a single calculator code instruction which will stack each of the strings "A" to "Z", without the need for CHR$. Just follow this instruction with "execute B" (code 3B) with B assigned with 1D, and the value of these single letter numeric variables will be placed on the calculator stack speedily and efficiently.

Also (perhaps) useful to know is the fact that the strings "B$", "W$", and "Z$" may be stacked in one single instruction. This means that if you store a string whose value is needed by machine code in either B$, W$ or Z$, then its value may be retrieved as easily as PI. Just stack the string (see Figure Eleven for instructions code) and follow the instruction with "execute B" (code 3B) with B containing 18. You'll find that this will save the computer an awful lot of work.

It is of course possible to stack any string in a single instruction, but to do this the text of the string must be stored separately somewhere in memory. See last month's article.

Using a BASIC array

There is one final, and very ingenious trick which we can learn. The trick is to use the calculator's memories in conjunction with a BASIC array. When we use the calculator instructions "store MO", "recal M3", et cetera, we are copying five byte numbers between the calculator stack and the memories. Usually there are six memories, which are stored amongst the system variables at address MEMBOT, but this isn't always the case. We can change the whereabouts of the calculator memories by altering the system variable (MEM). Note

10

```
061D     VAL_A      LD B,1D         B:= calc code for VAL.
EF                  RST 28          Switch calculator on.
3440B00041            stk data 41h  Stack code of "A".
2F                    chr$          Convert to string "A".
3B                    execute B     Convert to value of variable.
38                    end calc

0618     VAL_B$     LD B,18         B:= calc code for VAL$.
EF                  RST 28          Switch calculator on.
3440B00042            stk data 42h  Stack code of "B"
2F                    chr$          Convert to string "B".
3440B00024            stk data 24h  Stack code of "$".
2F                    chr$          Convert to string "$".
17                    s_add         Combine to form string "B$".
3B                    execute B     Convert to string value of var.
38                    end calc
```

11

To stack numeric variable names on calculator stack:

VARIABLE	CALCULATOR CODE	VARIABLE	CALCULATOR CODE
A	34B02B0201	N	34B00D0201
B	34B0050201	O	34B01F0201
C	34B0140201	P	34B0270201
D	34B01B0201	Q	34B02A0201
E	34B01A0201	R	34B0120201
F	34B0130201	S	34B0230201
G	34B00B0201	T	34B00A0201
H	34B0060201	U	34B00F0201
I	34B0170201	V	34B00C0201
J	34B00E0201	W	34B0220201
K	34B0160201	X	34B01C0201
L	34B01E0201	Y	34B0070201
M	34B0150201	Z	34B0240201

To stack string variable names on calculator stack:

VARIABLE	CALCULATOR CODE
B$	34B0B43E02
W$	34B08B2302
Z$	34B0BD3E02

12

```
10 DIM A(32)                          A() is first item in VARS area.
20 RANDOMIZE USR array                Execute machine code.


2A4B5C     ARRAY   LD HL,(VARS)       HL: points to BASIC array A().
010600             LD BC,0006
09                 ADD HL,BC          HL: points to first element of A().
22685C             LD (MEM),HL        Identify memories with elements.
EF                 RST 28             Switch calculator on.
3440B0002A           stk data 2A      Stack the number 2A.
CF                   store MF         Store in Memory OF.
02                   delete           Empty calculator stack.
38                   end calc         Switch calculator off.
21925C             LD HL,5C92,MEMBOT
22685C             LD (MEM),HL        Restore calculator memories.
C9                 RET
```

that this system variable always points to the start of memory zero. Memory one is at (MEM)+5, memory two at (MEM)+OA, and so on.

Figure Twelve contains some BASIC and some machine code. Line 10 creates a BASIC numeric array which, since it is the first variable encountered, will be the first variable in the VARS area. Line 20 runs the machine code which follows. The machine code alters the value of (MEM) so that is points to the variable A(1) in the VARS area. This has a rather stunning effect.

The array elements have now each acquired two different names, and each element may be accessed in two different ways. The firts element is known both as A(1) and MO. A(1) is the name used by BASIC, while MO is the name used by machine code. Similarly, A(2) is BASIC may be referred to in calculator code as M1, and so on up to A(32), which may now be referred to in calculator code as M1F. Notice that the number of available calculator memories has now increased from six to thirty two.

Any number of memories, up to a maximum of thirty two, is possible, and the number is determined by the dimension of A(). This now means that any reference to a calculator memory is in fact a direct reference to an array element. Such elements may be read — or even altered — by a machine code program.

The program of Figure Twelve will prove this, by assigning — from within machine code — the element A(16).

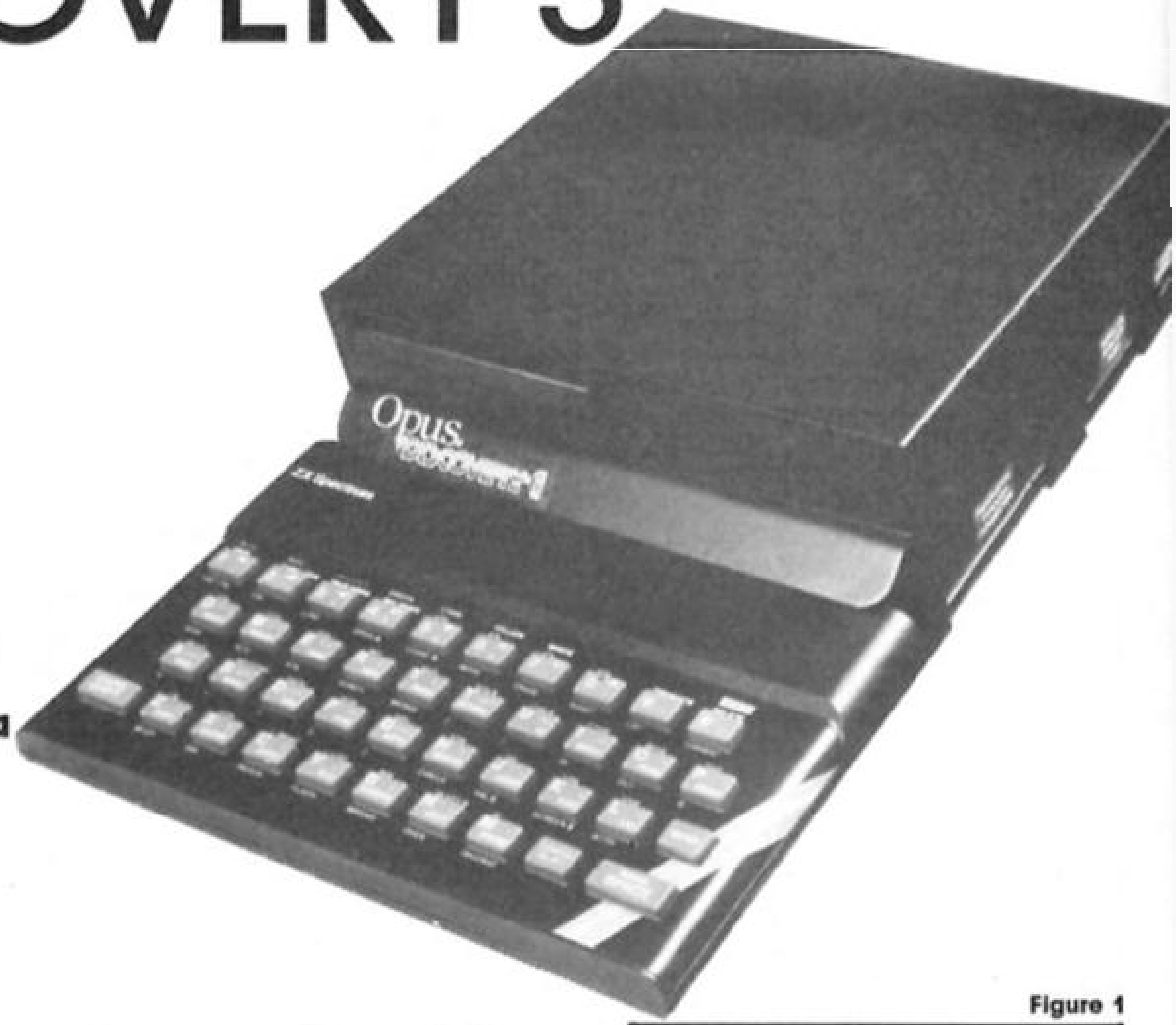
Note that the program restores (MEM) to its original value of MEMBOT before returning to BASIC. This should always be done. Note also that since A(1) to A(3) are equivalent to M0 to M2, they are liable to be corrupted by such functions as SIN etc (see Appendix in last month's article).

I think that's all there is to say for now on the subject of passing parameters between BASIC and machine code. I'll move on now to a new topic.

The function generator

There is just one calculator instruction left to cover. It is the function "series", otherwise known as the function generator, or series generator. Its code will be a value between 80h and 9F. The last five bits of the code form a parameter, so that 86 means "series 6", 8C means "series 12d", 99 means "series 25d", and so forth. This is the single most powerful instruction in the whole of the calculator set. It is the function with which SIN and EXP and others were written in the ROM. With it we may create our own designer functions, or implement mathematical functions which are not present (and not otherwise possible) on the Spectrum. The "series" function is the calculator's final frontier. To be continued....

MORE TRICKS WITH DISCOVERY'S CAT!



John Wase presents a round-up of useful routines to take full advantage of the Discovery disc drive.

This month we have got together a couple of routines which you might find useful in using the Discovery disc unit. Both these programs operate on Discovery's catalogue file. Regular readers will recall that this was introduced last month with a very simple demonstration program which read the file and then moved the recorded programs one by one from disc to disc; these programs are rather more sophisticated. The first is from Opus themselves; the second is from S. Nutting of Histon.

Full cat

This program gives more information about each file on the disc than merely cataloguing. It prints out the name of the file, together with the numbers of the start and finish sector (or block); readers will remember (from the Discovery manual, pages 10 and 11) that there are 18 sectors on each track and each sector holds 256 bytes so a forty track disc will hold up to 180 Kbytes (1 K being 1024 bytes). In addition, the program also prints out the number of bytes used in the last sector which has been written to: it is therefore very easy to calculate the precise number of bytes which a program takes up.

Programming points

As you type in the listing from figure 1, you will notice that this program is much more compact than the disc-to-disc routine published last month, which was written with the idea of making it absolutely clear how to access the catalogue file. However, the general principles are exactly the same. Much of the space is saved by defining a function in line 10 which can then be used later in the program. A typical printout from "full cat" is shown in figure 2. Note the third item printed in the array of three (number of bytes used in the last sector).

A quick glance at page 28 of the Opus manual confirms that in the CAT file, part of the first number (the first two bytes in the file) represents the number of bytes required minus 1; hence the addition of 1 in statement 60.

Supercat

This program reads "disc headers", getting its data from the catalogue file, and printing out a wealth of information in an attractive format. It is as good or better with disc headers as many a professional tape header reader. It consists of a 100 byte machine code routine

Figure 1

```
   5 REM "PROGRAM FULL CAT"
  10 DEF FN a()=CODE INKEY$#0+25
6*CODE INKEY$#0: DIM a$(10): DIM
 a(3)
  20 INPUT "drive:",d;#2;"filena
me","start end  last"''
  30 CLEAR #0: OPEN #0:" CAT ";d
 RND 16:POINT #0;1
  40 FOR j=1 TO 3: LET a(j)=FN a
(): NEXT j: FOR j=1 TO 10: LET a
$(j)=INKEY$#0: NEXT j
  50 IF a(3)=2^16-1 THEN STOP
  60 FOR j=1 TO 10: INVERSE 1: P
RINT "<"+STR$ CODE a$(j)+">" AND
 a$(j)<" ";: INVERSE 0: PRINT a$
(j) AND a$(j)>=" ";: NEXT j: PRI
NT TAB 16;a(2);TAB 22;a(3);TAB 2
7;a(1)+1
  70 GO TO 40
```

together with a bit of basic to run it and to print the files on the disc. The results are shown in figure 3. It prints the disc title: this is given the heading of "cartridge name". Then it gives the various program titles. Alongside the name of each program is the number of bytes. Then, on the same line, you get the start sector, the finish sector and the number of sectors used.

Underneath you are told if the program is a BASIC program, or if it is a numeric array, a character array, machine code or a print file. If it's BASIC, then you get the number of bytes for the program itself, together with the number of bytes for the variables and the line number from which the program autoruns. If it's code, then you get the familiar start and length.

filename	start	end	last
new	0	6	256
dumprog	7	9	41
dump	10	11	1
buff1CODE	12	13	7
ZEUS	14	40	151
assembler	41	43	185
code	44	44	43
catc	45	45	107
supercat	46	50	98
config	81	82	212
full cat	83	84	229
a	85	112	7

Figure 2

```
CARTRIDGE NAME =new
dumprog        546      7    9    3
Basic+Variables Basic Vari   Auto
BASIC:         546   518    28    1
dump           250     10   11    2
MACHINECODE 65118,250
buff1CODE      256     12   13    2
MACHINECODE 23296,256
ZEUS           6800    14   40   27
MACHINECODE 57344,6800
assembler      690     41   43    3
Basic+Variables Basic Vari   Auto
BASIC:         690   690     0    1
code           36      44   44    1
MACHINECODE 32768,36
catc           100     45   45    1
MACHINECODE 64900,100
supercat       1115    46   50    5
Basic+Variables Basic Vari   Auto
BASIC:         1115  1115    0 9999
a              6912    51   78   28
MACHINECODE 16384,6912
```

Figure 3

The program

The BASIC program is given in figure 4. This is typed in and saved with "SAVE*1;'supercat" LINE 9999. It contains the following variables: a points to the CAT area; i points to the start of the file information; b, s and e are the sector, its start and its end; type is the type of program (0 for BASIC, 1 for a numeric array, 2 for a character array, 3 for machine code and 4 for a print file); length is the program length; start is the start address if the program is machine code or the auto-start line if BASIC; finally, basic is the length of a BASIC program (without the variables).

```
   5 REM PROGRAM"SUPERCAT"
  10 INPUT "Drive:";d: POKE 6499
0,d
  20 RANDOMIZE USR 64900
  30 DEF FN a(x)=PEEK (x+a)+256*
PEEK (a+x+1)
  40 PRINT "CARTRIDGE NAME =";:
FOR a=62006 TO 62015: PRINT CHR$
 PEEK a;: NEXT a: PRINT
  50 LET a=62016: LET i=63800
  60 LET b=FN a(0)
  70 LET s=FN a(2)
  80 LET e=FN a(4)
  90 LET mem=(e-s)*256+b-6
 100 IF e=65535 THEN STOP
 110 FOR p=a+6 TO a+15: PRINT PA
PER 4;CHR$ PEEK p;: NEXT p
 120 PRINT mem;TAB 17;s;TAB 21;e
;TAB 25;e-s+1
 130 LET t=a: LET a=i: LET type=
PEEK a
 140 LET length=FN a(1)
 150 LET start=FN a(3)
 160 LET basic=FN a(5)
 170 IF type=0 THEN PRINT PAPER
1; INK 7;"Basic+Variables Basic
Vari Auto "'"BASIC:";TAB 10;leng
th;TAB 16;basic;TAB 22;length-ba
sic;TAB 27;start
 180 IF type=1 THEN LET n=PEEK (
i+4)-64: PRINT PAPER 3; INK 6;"N
UMERIC ARRAY    ";mem;TAB 21;" TY
PE=";CHR$ n,
 190 IF type=2 THEN LET c=PEEK (
i+4)-128: PRINT PAPER 3; INK 6;"
CHARACTER ARRAY ";mem;TAB 21;" T
YPE=";CHR$ c;"$",
 200 IF type=3 THEN PRINT PAPER
6; INK 2;"MACHINECODE ";start;",
";length,
 210 IF type>3 THEN PRINT PAPER
5;"PRINTFILE TYPE  LENGTH=";mem,
 220 LET a=t: LET a=a+16: LET i=
i+10: GO TO 60
9999 CLEAR 61999: PRINT "LOADING
 IN MACHINE CODE": LOAD *1;"catc
"CODE : RUN
```

Figure 4

```
                              ORG 64900
64900 205,008,023     start: CALL 5896       ;Call Discovery ROM in.
64903 006,000                LD B,0          ;Find from Discoverys Lookup Tables   the I/O Read & Write routine
64905 247                    RST #30         ;Then Load HL with the address.
64906 018                    DEFB 18
64907 034,226,253            LD (ADDRESS),HL ;POKE the Address in the Loadsect routine.
64910 033,000,000            LD HL,0         ;HL= Sector 0 (the start of the CAT info.
64913 017,048,242            LD DE,62000     ;Address to load CAT info.
64916 014,000          loop: LD C,0          ;256 bytes to load in.
64918 205,218,253            CALL loadsect   ;Load the Sector in.
64921 020                    INC D           ;Add 256 to current load CAT Address
64922 069                    LD B,L          ;B=Sector no.
64923 062,006                LD A,6          ;If sector no just loaded = 6 Then JR to loading the individual files in.
64925 184                    CP B
64926 040,003                JR Z,files
64928 035                    INC HL          ;If Sector <>6 Then load in next sector.
64929 024,241                JR loop
64931 033,046,249     files: LD HL,63790     ;Address to Load file data.
64934 221,033,050,242        LD IX,62002     ;Points to Filename info.
64938 017,010,000  nextfile: LD DE,10        ;Add 10 to current Load file data Address.
64941 025                    ADD HL,DE
64942 235                    EX DE,HL
64943 001,016,000            LD BC,16        ;Add 16 to current Filename data.
64946 221,009                ADD IX,BC
64948 221,229                PUSH IX
64950 213                    PUSH DE
64951 221,078,002            LD C,(IX+2)     ;If BC=65535 then we have loaded all files in.
64954 221,070,003            LD B,(IX+3)
64957 033,255,255            LD HL,65535
64960 237,066                SBC HL,BC
64962 040,016                JR Z,end        ;RET by JR to end If BC=65535
64964 221,110,000            LD L,(IX+0)     ;HL=Sector no to load.
64967 221,102,001            LD H,(IX+1)
64970 014,007                LD C,7          ;Load in 7 bytes.
64972 205,218,253            CALL loadsect   ;Do actual loading.
64975 225                    POP HL
64976 221,225                POP IX
64978 024,214                JR nextfile     ;Loop until loaded all data
64980 225               end: POP HL
64981 221,225                POP IX
64983 195,072,023            JP 5960         ;CALL Sinclair ROM in and RET to Basic.
64986 197          loadsect: PUSH BC
64987 213                    PUSH DE
64988 229                    PUSH HL
64989 062,001                LD A,1          ;Drive 1 (optional If A=2 then load from Drive 2).
64991 006,002                LD B,2          ;Used by Discovery ROM.
64993 205                    DEFB 205        ;When I/O Address is found the OP Code would turn out to be CALL nn.
64994 000,000                DEFW 0
64996 225                    POP HL
64997 209                    POP DE
64998 193                    POP BC
64999 201                    RET
```

Figure 5

Machine code

The next section contains an assembly language listing of the machine code (figure 5). The comments alongside it are pretty self explanatory. Thus, CALL 5896 pages in the Discovery ROM and LD B, 0 finds from Discovery's ROM the LOAD address routine: this is then POKED into the loadsector routine. The sector number, address to load the sector at and the number of bytes to load (0=256 bytes) then follow.

Then there is the option of loading a sector (LD B, 2) or saving on disc (LD B, 0); the drive number to load from being loaded into A. The program then loops round to load the next file data, and so on, until it finally pages in the Sinclair ROM and returns to BASIC.

```
   5 REM "TO ENTER CODE"
  10 CLEAR 64899: LET c=0: FOR a
=64900 TO 64999: READ n: POKE a,
n: LET c=c+n: NEXT a
  20 IF c<>11267 THEN PRINT "Dat
a Error"
  30 STOP
  40 DATA 205,8,23,6,0,247,.....
.........etc, etc.........201 - ent
er all the data from the assembl
y language program, second colum
n.
```

Figure 6

Basic loader

For those of us without an assembler, there is a BASIC program (figure 6) to POKE in the decimal numbers. These are listed alongside the assembly language listing, in the next column to the list of addresses. If you intend to use this program, type it in carefully, amending line 40 so that it contains *all* the decimal numbers in the second column of the assembly listing *in the order in which they are printed.* Run it and save the code which has been created with SAVE*1";"catc"CODE 64900, 100. It will keep all those programs (you know, the ones you recorded last year) tidily sorted out, so that you know what they are!

And finally. . .

The Opus drive is now, indisputably, the most commonly found and the most popular. If you have a short routine which you think merits publication, please send it to the Editor, ZX Computing, marked "Opus program". We could then, perhaps, make this column a regular feature.

QL COLUMN

Brian Beckett reviews Super Media Manager.

I hate to say "I told you so" (but I did) and all is still quiet on the Amstrad Front despite last month's flurry of excitement over a quick deal with the newly-formed QL Suppliers Group. Avid Beckett (the other one) readers will also be glad to know that Godot would seem to be alive and well and working for Amstrad's PR firm as I'm still waiting for him to "check with the client and get back to me" regarding last month's quest for a quote on the QL's future (or lack of it) so I could pass it on to an information-starved public.

This may actually be a rare case of "no news" indeed being "good news". There's little doubt that Amstrad would like to see the QL quietly sink into oblivion but so far its supporters have kept it afloat by steadfastly refusing to accept defeat. Software houses and peripheral manufacturers have invested too much money in QL products to simply give up without a fight and the computer's 100,000 plus owners are just about enough of a market to make the fight worthwhile in terms of the balance sheet. US interests have joined in the effort to acquire rights to Sir Clive's brainchild and — as long as the talking goes on — there is a very real possibility of some sort of agreement soon that will keep it alive in spite of everything. Which ever way things go, however, I can look forward to a quote from Amstrad's PR man — thereby successfully ending an epic adventure that is beginning to rival Galahad's search for the Grail.

Managing Media

As promised, my **Super Media Manager** from Digital Precision arrived a couple of weeks ago and looks to be just about everything that the company claims for it. The folks at Digital have many virtues but understating things is not one of them so when one of their packages matches all the superlatives (and the ones I've played with do) you know that you've got your hands on a winner. It also sells for £39.95 which — if not actually cheap — is commendably reasonable considering what you're getting for the money and if you're feeling a bit flush you can save a few bob by buying **Supercharge** (£59.95) at the same time and get Super Media Manager (I do wish Digital would dig up a different superlative and do style-conscious writers a good turn) for a tenner less.

Super Media Manager comes on either a single disc or four microdrives and is accompanied by a well-presented manual which is both informative and deliberately humorous. Generally the program allows you to do just about anything you might wish with microdrive or disc files and to copy between them with more flexibility than I've seen in competing products (of which there aren't all that many anyway). The program is extensively menu driven, screen displays are good and the commands are generally easy to follow and operate. Facilities exist for extensive sector editing and copying (disc and microdrive) and for recovering lost or damaged files as long as there is something there for the Manager to locate.

I've used it to recover some stuff on dodgy microdrives (so far my disc package has behaved decently enough) and was most impressed with the results. The stuff I went after wasn't important but the facility to get "lost" data back again is almost a prime directive — especially if you're solely dependent upon microdrives which, if they aren't nearly as unreliable as their critics claim, are temperamental to say the very least.

The Manager's most impressive feature is probably its ability to read and copy data from non-QL discs and, in particular, from the IBM and IBM-compatible MS-DOS system. I really haven't had a chance to explore the full range of possibilities opened up by this option but a couple of quick test runs lived up to Digital's promise without fault. BBC and CP/M systems may also be read or copied. I don't think I need to dwell on all the implications offered by this facility and the benefits arising but suffice it to say that it gives the QL a powerful boost and goes a long way towards bringing the QL into the mainstream micro club. The Manager will not be fully compatible with all QL disc interfaces (those which have not been manufactured to include all the facilities of the QL's disc operating ROM) and Digital suggest that — if there are any doubts — that would-be buyers check up on their systems beforehand. For what it's worth, the Manager is not protected by the Lenslok security system (unlike Supercharge) which I find a blessing and I hope that Digital is rewarded with sufficient profits to compensate for any successful piracy that might arise.

QL Bargains

By the time you read this, the PCW Show will have come and gone and we will probably know whether or not the QL is to survive in anything like its present form. We should also have a pretty good idea if any of the various "sons-of-QL" schemes being bandied about are going to get off the ground. Either way you — unless you've decided to consign your beloved QL to the outer darkness and are busy looking at IBM clones — should keep a sharp eye out for peripheral bargains.

A lot of companies have invested in QL-specific peripherals and a lot of them won't survive the competition even if the computer lives on. The most obvious is memory expansion cards, modules combined RAM/disc interfaces and so on. Just about everybody involved in the QL business has some sort of memory add-on available and — taking Darwin to heart — a lot of them wouldn't make it even in a healthy market.

This is not a criticism of the various products (although there are a lot of criticisms one could make in some cases) but just a recognition of economic reality. But the market isn't all that healthy and there's a lot of excess competition in a few areas so, if you're in the mood for a bit of expansion, shop around for a good deal because you will probably find it. Sadly (from the buyer's point of view) this doesn't yet apply to software but — if the QL does go the way of all flesh — the time will certainly come.

To close on a positive note, Saga Systems is releasing a QL version of its remote control 2001 keyboard for the Spectrum. It's about time somebody gave us an attractive, alternate keyboard but (damn it) it doesn't match my disc interface. If it accommodates yours without a hassle, I'd have a good look as Saga's Spectrum keyboards are very good.

SPECTRUM/QL LINK-UP

David Nowotnik examines the possibilities for network communications between Spectrum and QL.

One of the major growth areas for the computer industry is communications — linking one computer with another. This involves the movement of electronic information between computers. distances may be a matter of a few centimetres, or half way round the world. Today, it is considered trivial for computer users in the UK to be 'talking' to computers in Europe and the USA, accessing business and scientific databases in a matter of seconds.

One of the main problems in supporting this communications growth is the incompatibility between different computers. All sorts of electronic 'black boxes' are employed to allow two computers to communicate when they were never designed to allow such a link.

For microcomputers, communication is normally quite simple. With the RS232 standard adopted by most manufacturers, many microcomputers are able to talk to each other. Although they are quite different micros, the ZX Spectrum (with Interface 1) and QL are fitted with RS232 ports. Linking the two micros with a specially constructed lead, and setting equal baud rates, parity bits, etc, will allow data and programs to be exchanged between these micros.

But the Spectrum (via its Interface 1) and QL have another option for communication, which quite simply involves joining both computers with the lead fitted with two mini jack plugs supplied by the manufacturer — the network.

In both QL and Spectrum, the network was designed for 'local area' connection of two or more computers. While the specific BASIC commands which operate the network have differences, the method of operation, and the communication standards are exactly the same. Networking between groups of QLs or Spectrums alone is very straightforward, the manuals of both computers carry examples. Networking in systems containing a mixture of the two micros is not quite so simple, but applying a few simple rules, it can be made to work reasonably well.

There can be many reasons for wanting to link home computers. In this article, we'll be examining just two; passing a BASIC program from Spectrum to QL, and using the Spectrum to drive an extra screen for the QL. With these examples the ground rules for communication will be established, allowing you to develop whichever communication application you have in mind.

Fig.1. Simple example of listing a Spectrum program on the QL

a) Spectrum listing

```
100 REM Example of LIST
110 REM
120 FORMAT "n";2
130 REM
140 REM Ready to send
150 REM
160 OPEN # 4;"n";1
170 INPUT #4;a$
180 CLOSE # 4:  PRINT a$
190 REM
200 REM Now send Program
210 REM
220 OPEN # 5;"n";1
230 LIST #5
240 CLOSE # 5
```

b) QL Listing

```
100 REMark Example of LIST
110 :
120 NET 1
130 :
140 REMark ready to receive
150 :
160 OPEN #4,neto_2
170 PRINT #4;"XXX"; CHR$(13);
180 CLOSE #4
190 :
200 REMark Now receive program
210 :
220 COPY neti_2 TO scr
```

Transfer requests

There must be many Spectrum owners who have upgraded to the QL, then would like to transfer BASIC programs from their old micro to the 68008 machine. This usually meant totally re-typing the program. However, if both the Spectrum (plus Interface 1) and QL are available, then transfer of programs via the network could be possible. Few transferred programs will work without some modification, but at least a lot of typing can be saved by a relatively simple electronic transfer.

Looking at the manuals of the two micros, simple LOAD/SAVE instructions will transfer programs between micros of the same type. But such simplicity is not available to Spectrum-QL transfers. The QL saves and loads its SuperBASIC programs as ASCII files; what you see when you list the program is exactly what you get when the program is saved on microdrive. With the Spectrum, the way the program is arranged in memory is the way it appears on tape, with line numbers condensed to two bytes, keywords tokenised, and numbers within the program always appearing in two forms. So, to transfer a program from Spectrum to QL, a translation is needed.

At first glance, the simplest method of transfer would appear to be the LIST command. Whenever it is used on the Spectrum, to screen, printer, or via the RS232 line, this command appears to carry out all translations necessary to convert the tokenised version of BASIC stored in the Spectrum to an ASCII listing of a BASIC program. But not so on the network. To demonstrate this, try the simple example in fig.1. With the program lines typed into both QL and Spectrum, and the network lead connecting both computers, enter RUN on both (the order, or precise timing does not matter; the Spectrum will not broadcast until the QL is listening, and vice versa). Note that, on the Spectrum, 'OPEN# ' and 'CLOSE# ' are individual keywords, quite distinct from 'OPEN' and 'CLOSE', whereas 'INPUT# ' has to be entered as

'INPUT' followed by '#'.

What you should see is 'XXX' on the top of the Spectrum screen, and, after a pause of a few seconds, something vaguely close to a listing of the Spectrum program on the QL screen.

Station to Station

Line 120 in both listings in fig.1. sets up the station number of the Spectrum (FORMAT "n";2) and QL (NET 1) as 2 and 1 respectively. Network station numbers are used primarily when more than two computers are linked into the network, so that each micro can be identified uniquely. Both micros have a default network station number of one, and it is possible when only two micros are in the network for them both to take the default value. But in all the examples in this article, the Spectrum is assigned the nurr ber 2, and QL 1.

In lines 160 to 180, the QL sends (and the spectrum receives) a message to say that both are ready. This isn't strictly necessary, as the broadcasting station always waits for the receiver to listen, except in a general broadcast (to several receiver micros). But these program lines do serve to show you some of the rules of communication.

In the Spectrum listing, line 160 opens channel 4 for network communication to the micro with a station number of 1. In opening a network channel, the Spectrum is not concerned at the direction of communication. Not so with the QL. In line 160 of the QL listing, channel 4 is opened specifically for transmission (neto) to micro number 2. 'neti' is the device name to be used on the QL for receiving transmissions (as in line 220, 'neti_2' commands the receipt of a transmission from station 2).

The first communication between the two micros occurs in line 160 in both listings. The Spectrum receives a stream of bytes, and places them in the string 'a$'. Those bytes are 'XXX', followed by character code 13. The latter is the end of line marker for the Spectrum. The corresponding end of line character for the QL is character code 10. Forgetting that incompatibility can cause quite a few problems! Normally, the computer will add the end of line character to any PRINT stream, unless specifically told not to by the PRINT statement. For example, in line 160 of the QL listing, the end ; prevents the QL end of line character being sent.

In all network communications between Spectrum and QL using the PRINT command, the semi-colon is the best spacer to be used. Particularly in Spectrum to QL communications, use of commas (e.g. PRINT # 4,a$,b$, CHR$ 10) will cause extra control characters to be inserted; in this example, character code 6 for each comma encountered, and character code 13 at the end of the print stream as default, as no semi-colon has been added.

Spectrum lines 220 to 240 send the program listing down the network to station 1. The simple COPY command (line 220, QL listing in fig.1.) takes whatever is presented on the network and dumps it on the screen. What you'll see are a continuous stream of characters. The Spectrum LIST command has translated the line numbers, but not the keywords. And the Spectrum's end of line code is not currently interpreted by the QL, so no new lines occur at the end of BASIC lines. So, to use LIST on the Spectrum requires further translation; the listings in fig.2. are routines to effect that translation.

Reservations

The short Spectrum program in fig.2. should be MERGEd with any Spectrum program you wish to transfer, then the copy routine initiated with the direct command 'RUN 9990'. The QL program in fig.2 works by reserving space at the top of RAM (line 170) and creating an array of the keywords (lines 200-250). The Spectrum is told to start listing (lines 290-320), and the bytes received via the

Fig.2. Transfer a Spectrum program to QL using LIST

a) Spectrum listing

```
9990 REM program copy routine
9991 FORMAT "n";2
9995 OPEN # 4,"n",1: INPUT #4;a$: CLOSE # 4
9996 IF a$="s" THEN  STOP
9997 OPEN # 5,"n",1: LIST #5: CLOSE # 5
9998 GO TO 9995
9999 STOP
```

b) QL Listing

```
100 REMark Import Spectrum Programs
110 REMark via the network system
120 :
130 REMark Initialise
140 :
150 CLS :PRINT " Initialising - please wait"
160 a= RESPR (0)
170 IF a> 221184 THEN a= RESPR (a-221184)
180 NET 1
190 z$= CHR$(13)
200 position=a: RESTORE 1000
210 DIM keyword$ (91,10), key_len (91)
220 FOR i=1 TO 91
230   READ a$: keyword$ (i)=a$
240   key_len (i)= LEN (a$)
250 END FOR i
260 :
270 REMark Transfer program from Spectrum
280 :
290 OPEN #4, neto_2
300 PRINT #4;"g";z$
310 CLOSE #4
320 OPEN #5, neti_2
330 PRINT \" Receiving the program"\" from the Spectrum"
340 REPeat loop1
350   IF EOF (#5) THEN EXIT loop1
360   x$= INKEY$(#5,0)
370   POKE position, CODE (x$)
380   position=position+1
390 END REPeat loop1
400 CLOSE #5
410 PRINT \" Program transfer complete"\\
420 OPEN #4, neto_2
430 PRINT #4;"s";z$
440 CLOSE #4
450 :
600 REMark Create file of BASIC lines
610 :
620 PRINT " Place a formatted cartridge"
630 PRINT " in mdv2_, then press any key"\\
640 x$= INKEY$(-1)
650 DELETE mdv2_temp_bas
660 OPEN_NEW #4,mdv2_temp_bas
670 REPeat loop2
680   a$=""
690   REPeat loop3
700     z= PEEK (a)
710     SELect ON z
720     =32 TO 127: a$=a$& CHR$(z)
730     =128 TO 143: a$=a$&'()'
740     =144 TO 164: a$=a$& '('&CHR$(z-79)&')'
750     =165 TO 255
760       x$=keyword$ (z-164): x$=x$(1 TO key_len (z-164))
770       a$=a$&x$
780     =13: EXIT loop3
790     END SELect
800     a=a+1
810   END REPeat loop3
820   a=a+1
830   PRINT #4;a$
840   PRINT a$
850   IF a>=position THEN EXIT loop2
860 END REPeat loop2
870 CLOSE #4
1000 DATA ' RND ',' INKEY$ ',' PI ',' FN '
1010 DATA ' POINT ',' SCREEN$ ',' ATTR ',' AT '
1020 DATA ' TAB ',' VAL$ ',' CODE ',' VAL '
1030 DATA ' LEN ',' SIN ',' COS ',' TAN '
1040 DATA ' ASN ',' ACS ',' ATN ',' LN '
1050 DATA ' EXP ',' INT ',' SQR ',' SGN '
1060 DATA ' ABS ',' PEEK ',' IN ',' USR '
1070 DATA ' STR$ ',' CHR$ ',' NOT ',' BIN '
1080 DATA ' OR ',' AND ',' <= ',' >= '
1090 DATA ' <> ',' LINE ',' THEN ',' TO '
1100 DATA ' STEP ',' DEF FN ',' CAT ',' FORMAT '
1110 DATA ' MOVE ',' ERASE ',' OPEN # ',' CLOSE # '
1120 DATA ' MERGE ',' VERIFY ',' BEEP ',' CIRCLE '
1130 DATA ' INK ',' PAPER ',' FLASH ',' BRIGHT '
1140 DATA ' INVERSE ',' OVER ',' OUT ',' LPRINT '
1150 DATA ' LLIST ',' STOP ',' READ ',' DATA '
1160 DATA ' RESTORE ',' NEW ',' BORDER ',' CONTINUE '
1170 DATA ' DIM ',' REM ',' FOR ',' GO TO '
1180 DATA ' GO SUB ',' INPUT ',' LOAD ',' LIST '
1190 DATA ' LET ',' PAUSE ',' NEXT ',' POKE '
1200 DATA ' PRINT ',' PLOT ',' RUN ',' SAVE '
1210 DATA ' RANDOMIZE ',' IF ',' CLS ',' DRAW '
1220 DATA ' CLEAR ',' RETURN ',' COPY '
```

Fig.3. Create Spectrum Keywords as QL DATA statements

a) Spectrum listing

```
10 FORMAT "n",2
20 REM Wait for start signal
30 OPEN # 4,"n",1
40 INPUT #4;a$
50 CLOSE # 4
60 IF a$ <> "start" THEN  GO TO 30
70 REM now start to send the data
80 LET x=0
90 LET y= PEEK x
100 OPEN # 5,"n",1
110 PRINT #5; CHR$ y;
120 CLOSE # 5
130 LET x=x+1
140 IF x>16383 THEN  STOP
150 REM Wait for instruction
160 REM       to proceed
170 OPEN # 4,"n",1
180 INPUT #4;a$
190 CLOSE # 4
200 IF a$="g" THEN  GO TO 90
210 STOP
```

b) QL Listing

```
100 REMark  Create a list of Spectrum Keywords
110 REMark        Initialise
120 NET 1
130 DIM line$ (23,60),line_len (23)
140 z$= CHR$ (13): a$="   ": b$="RN"&CHR$(196)
150 REMark Send signal to start search
160 OPEN #4, neto_2: PRINT #4;"start";z$;: CLOSE #4
170 REMark Now collect bytes
180 PRINT " Searching Spectrum ROM"
190 REPeat loop1
200   OPEN #4, neti_2
210   y$= INKEY$(#4,-1)
220   CLOSE #4
230   a$=a$&y$: a$=a$(2 TO 4)
240   IF a$=b$ THEN EXIT loop1
250   send_next
260 END REPeat loop1
270 REMark Now found the start of the command
280 REMark table load in that table, creating
290 REMark new SuperBASIC DATA statements
300 line_no=1000: item=1
310 PRINT \" Found start of keywords"\
320 REPeat loop2
330   a$=line_no&" DATA "
340   IF line_no=1000 THEN a$=a$&"' RND ',"
350   REPeat loop3
360     b$=""
370     REPeat loop4
380       send_next
390       OPEN #4, neti_2
400       y$=INKEY$(#4,-1)
410       CLOSE #4
420       IF CODE (y$)>128 THEN
430         b$=b$&CHR$(CODE (y$)-128)
440         EXIT loop4
450       ELSE : b$=b$&y$
460       END IF
470     END REPeat loop4
480     a$=a$&"' "&b$&" '"
490     item=item+1
500     IF item>3 THEN item=0: EXIT loop3
510     IF line_no=1220 AND item=3 THEN EXIT loop3
520     a$=a$&","
530   END REPeat loop3
540   x=(line_no-1000)/10+1
550   line$(x)=a$: line_len(x)=LEN (a$)
560   line_no=line_no+10: PRINT a$
570   IF line_no>1220 THEN EXIT loop2
580 END REPeat loop2
590 REMark Instruct Spectrum to stop
600 OPEN #4, neto_2: PRINT #4;"s";z$;: CLOSE #4
610 REMark now create an ASCII file on mdv2
620 OPEN_NEW #5, mdv2_data_bas
630 FOR i=1 TO 23
640   a$=line$(i): a$=a$(1 TO line_len(i))
650   PRINT #5;a$
660 END FOR i
670 CLOSE #5
680 STOP
690 DEFine PROCedure send_next
700 OPEN #4, neto_2
710 PRINT #4;"g";z$;
720 CLOSE #4
730 END DEFine send_next
```

network are stored sequentially in the reserved area of RAM (lines 340-390). Translation of the BASIC keywords takes place via the routines of 600 to 870; the program lines are displayed on the screen, and are saved onto microdrive 2 under the file name 'temp_bas'.

The data lines (1000 onwards) could be typed in as listed, although all these keywords exist in the Spectrum ROM, so it is possible to use the network again, to transfer these keywords across, then create the DATA lines automatically. This is achieved using the program in Fig.3. Of course, by the time you have typed in all the program lines in fig.3., you could have typed in the DATA lines two or three times over! But Fig.3. does contain some more networking principles, and shows you how to create SuperBASIC lines automatically.

The Spectrum program in Fig.3. causes the Spectrum to become the 'slave' of the QL, obeying instructions passed down the network. The routine PEEKs sequentially bytes from the Spectrum ROM until told to stop by the QL. In the meantime, the QL is searching for the start of the keyword table in the Spectrum ROM (lines 190-260). This search is necessary as the keyword table is stored in a different place in different versions of the Spectrum ROM.

Once the first keyword (RND) is found, then the other 90 keywords are read in. The keywords are stored in the Spectrum ROM sequentially, with no gaps between each keyword. The end of a keyword is marked by the value 128 being added to the code value of the last character, and this is spotted in line 420. SuperBASIC lines are built up in the string a$, with four keywords per line (line 500). Each line is stored in the array line$ and its length recorded in array line_len (line 550). Once all keywords have been collected, the 23 DATA lines are written to a file 'data_bas' on microdrive 2 (lines 610 to 670). The command 'LOAD mdv2_data_bas' will load these SuperBASIC lines, to which the others in fig.2. can be added.

And now, back to fig.2., and the program transfer. As the QL doesn't have block graphics characters, or accessible user defined graphics, the transferred codes for these (128 to 164) are translated as '{ }' for block graphics, and '{ letter }' for udgs, where 'letter' corresponds to the letter defining the udg in question. Otherwise, the translation should correspond closely with the program as LISTed on the Spectrum.

Error terror

One thing you may notice as the transferred program is listed on the QL screen is that the number of transfer errors is quite high. Unfortunately, networking between Spectrum and QL can be prone to an unacceptably high number of errors. This seems to occur when a large number of bytes are transferred at once. In LISTing the program to its network port, the Spectrum collects 256 bytes in a buffer first; only when the buffer is full is the data sent down the line. Transfer of this cascade of data is satisfactory between two Spectrums, or two QLs, but, apparently not between Spectrum and QL.

Waiting for the buffer to fill before sending data also occurs when using PRINT to transmit information. So, you won't always get an immediate response from the receiving computer sending data down the network system. However, if the transmitter network channel is closed, then the buffer is emptied, and any data in it transmitted, no matter how full (or empty) the buffer happens to be. So to send small packets of data (which does seem to be reliable on Spectrum/QL network transfers) is possible by OPENing and CLOSEing channels immediately before and after PRINTing data to the channel. And this is the principle adopted by the programs in fig.4., which reliably transfers spectrum programs to the QL.

The Spectrum program in fig.4 measures the values of the PROG and VARS system variables (line 9981-9982), their difference being the length of the BASIC program. This information is passed to the QL (line 9984). The Spectrum then passes the BASIC program to the QL, byte by byte, until instructed to stop (lines 9991

to 9994). Before PEEKing each byte, PROG is measured again (line 9986), as the BASIC program can move about in RAM while the network is in operation.

In typing in the QL program in fig.4., remember to add the DATA lines (1000-1220) in fig.2., to permit the translation of keyword codes. This QL program works in much the same way as the program in fig.2., except that more translation is necessary; to extract the line number (line 690) and to deal with the way numbers are stored in the Spectrum (line 810). Line 700 prevents the Spectrum transfer routine from appearing on the 'temp_bas' file on microdrive 2; delete line 700 if you want to transfer this routine.

As before, MERGE the Spectrum program lines of fig.4a. with the program you wish to transfer, then type in 'RUN 9980', and RUN the QL program. Transfer is *much* slower than the LIST method used in fig.2., but errors are rare. In preparation of this article, most of the Spectrum programs were transferred to the QL using this routine, then imported into Quill, to produce what you seen in the figures.

Once the file 'temp_bas' has been created, it can be loaded into the QL RAM, and treated exactly as a QL program. Where the QL finds a mistake in syntax (where there are differences between Spectrum and QL), the QL adds the word 'Mistake' just after the line number, to help you spot at least some of the changes which are necessary to get the program to run correctly.

Variable numbers

Within the listings in fig.4. is an example of how numeric variables can be transferred from Spectrum to QL. In line 9983 of the Spectrum program, the variable 'k' takes the length (in bytes) of the BASIC program in the Spectrum's memory. This is transmitted to the QL in line 9984 by converting to string (STR$) and PRINTing the string. At the QL end, the number is received as a string and converted back to a numeric variable (line 300). The corresponding sequence for the transfer from QL to Spectrum would be:

QL:

200 num$=num: OPEN 5, neto_2
210 PRINT 5;num$; CHR$ (13);
220 CLOSE 5

Spectrum:

200 OPEN 5;"n";1
210 INPUT z$: z= VAL z$
220 CLOSE 5

In this way, numbers can be transferred between the two computers. This might have some application if you had a lot of number crunching to do; one computer could get on with that while the other was performing some other tasks.

The final example of a networking application is in the use of the Spectrum to provide a second screen for the QL. Despite the QL window facility, it's sometimes much more convenient to have two screens. Professional micros such as the IBM PC have this capability. If, for example, you are constructing a complex diagram on the QL with some computer aided design package, and want a 'HELP' message, you don't really want much of your diagram to be obscured by the help window. Far better to have those help messages appear on another screen.

The example in fig.5. is a simple way of achieving this. The QL sends out instructions via the network; pressing one of the function keys produces a help screen on the Spectrum, leaving the QL screen untouched. Each help screen has a number of pages; the total for each is listed in the DATA line 1010 of the Spectrum listing. For example, help screen 2 (obtained by pressing 'F2' on the QL) has three pages.

Switching from one page to another is effected by pressing the appropriate function key a second, and third time. The example help pages carry some of the important principles highlighted in this article. Line 1230 of the Spectrum listing directs the program to the appropriate set of print messages for any one help page. You could add/subtract your own set of messages for your own application.

The Spectrum is operating as a 'slave' to the QL; apart from typing in (or loading) the help print statements and network routines in fig.5a., there is no need to touch the Spectrum keyboard; the Spectrum receives

Fig.4. Alternative (slow but sure) transfer of Spectrum programs to QL

a) Spectrum listing

```
9980 FORMAT "n";2
9981 LET x= PEEK 23635+256* PEEK 23636
9982 LET y= PEEK 23627+256* PEEK 23628
9983 LET k=y-x: LET r=0
9984 OPEN # 5;"n";1: PRINT #5; STR$ k; CHR$ 10;: CLOSE #5
9985 OPEN # 5;"n";1
9986 LET x= PEEK 23635+256* PEEK 23636
9987 LET y= PEEK 23627+256* PEEK 23628
9988 LET z$= CHR$ ( PEEK (x+r))
9989 PRINT #5;z$;
9990 LET r=r+1: CLOSE # 5
9991 OPEN # 4;"n";1
9992 INPUT #4;a$
9993 CLOSE # 4
9994 IF a$="s" THEN  STOP
9995 GO TO 9985
```

b) QL Listing

```
100 REMark Import Spectrum Programs
110 REMark via the network system
120 :
130 REMark Initialise
140 :
150 CLS :PRINT " Initialising - please wait"
160 a= RESPR (0)
170 IF a> 221184 THEN a= RESPR (a-221184)
180 NET 1
190 z$= CHR$(13)
200 position=a: RESTORE 1000
210 DIM keyword$ (91,10), key_len (91)
220 FOR i=1 TO 91
230   READ a$: keyword$ (i)=a$
240   key_len (i)= LEN (a$)
250 END FOR i
260 :
270 REMark Transfer program from Spectrum
280 :
290 OPEN #5, neti_2
300 INPUT #5;num$: num=num$
310 CLOSE #5
320 PRINT \" Receiving the program"\" from the Spectrum"
330 FOR i=1 TO num
340   OPEN #5, neti_2
350   x$= INKEY$(#5,-1)
360   POKE position, CODE (x$)
370   position=position+1
380   CLOSE #5
390   OPEN #4, neto_2
400   IF i<num THEN PRINT #4;"g";CHR$(13);
410   IF i=num THEN PRINT #4;"s";CHR$(13);
420   CLOSE #4
430 END FOR i
440 PRINT \\" Transfer complete"\\
450 :
600 REMark Create file of BASIC lines
610 :
620 PRINT " Place a formatted cartridge"
630 PRINT " in mdv2_, then press any key"\\
640 x$= INKEY$(-1)
650 DELETE mdv2_temp_bas
660 OPEN_NEW #4,mdv2_temp_bas
670 REPeat loop2
680   REMark line number first
690   a$=256*PEEK(a)+PEEK (a+1): a=a+4
700   IF a$="9980" THEN EXIT loop2
710   REPeat loop3
720     z= PEEK (a)
730     SELect ON z
740     =32 TO 127: a$=a$& CHR$(z)
750     =128 TO 143: a$=a$&'{}'
760     =144 TO 164: a$=a$& '{'&CHR$(z-79)&'}'
770     =165 TO 255
780       x$=keyword$ (z-164): x$=x$(1 TO key_len (z-164))
790       a$=a$&x$
800     =13: EXIT loop3
810     =REMAINDER : a=a+5
820     END SELect
830     a=a+1
840   END REPeat loop3
850   PRINT #4; a$
860   PRINT a$
870   a=a+1: IF a>=position THEN EXIT loop2
880 END REPeat loop2
890 CLOSE #4
```

all its instructions from the QL.

While the network can connect QL and Spectrum very easily with the manufacturer's lead, we've seen here that it isn't all that easy to transfer data and programs reliably and quickly. The RS232 ports are much better in this respect, and we'll be examining this method of QL to Spectrum communication within the next few months.

Fig.5. Example of the use of the Spectrum to provide a 'slave' screen giving help messages for the QL

a) Spectrum listing

```
1000 FORMAT "n";2
1010 DATA 2,3,1,1,1
1020 DIM a(5)
1030 RESTORE
1040 FOR i=1 TO 5
1060 READ a(i)
1070 NEXT i
1080 LET page=0
1090 LET help=0
1100 OPEN # 4;"n";1
1110 INPUT #4;z#
1120 CLOSE # 4
1130 LET z= CODE z#
1140 IF z<1 OR z>5 THEN  GO TO 1100
1150 IF z <> help THEN  GO TO 1200
1160 LET page=page+1
1170 IF page>a(z) THEN  LET page=1
1180 GO TO 1220
1200 LET help=z: LET page=1
1220 CLS : PRINT  AT 2,10;"Help ";help; AT 4,10;"Page ";page;
AT 6,0;
1230 GO SUB 1000*(1+help)+100*(page-1)
1240 PAUSE 500: GO TO 1100
2000 REM Help 1; page 1
2010 PRINT "Remember to send only small     chuncks of data
via the network."
2090 RETURN
2100 REM Help 1; page 2
2110 PRINT "Use semi-colons as separators   whenever
possible."
2190 RETURN
3000 REM Help 2; page 1
3010 PRINT "NET is the command to assign    station on the
QL."
3020 PRINT : PRINT "FORMAT is used on the Spectrum"
3090 RETURN
3100 REM Help 2; page 2
3110 PRINT "OPEN # and CLOSE # are quite    different
keywords to OPEN and  CLOSE on the SPECTRUM."
3190 RETURN
3200 REM Help 2; page 3
3210 PRINT "INPUT, PRINT, and INKEY# are    the transmission
and receiver"
3220 PRINT "commands on both QLand Spectrum.They are used
with # and the    channel number"
3290 RETURN

4000 REM Help 3; page 1
4010 PRINT "CHR# 13 is the end of line      character on the
Spectrum."
4090 RETURN
5000 REM Help 4; page 1
5010 PRINT "CHR# 10 is the end of line      character on the
QL."
5090 RETURN
6000 REM Help 5; page 1
6010 PRINT "Numbers can be transferred via  the network by
converting to    strings."
6020 PRINT "On the Spectrum VAL converts thestring back to a
number."
6090 RETURN
```

b) QL Listing

```
10 REMark  Help procedure
20 REMark  producing help pages
30 REMark  on the Spectrum screen
40 :
50 REPeat loop
60   help
70   PAUSE 200
80 END REPeat loop
90 :
1000 DEFine PROCedure help
1010 LOCal a,z
1020 z=KEYROW(0)
1030 IF z=0 OR z>32 OR z=4 THEN RETurn
1040 SELect ON z
1050    =1: a=4
1060    =2: a=1
1070    =8: a=2
1080    =16:a=3
1090    =32:a=5
1091    = REMAINDER : a=0
1092 END SELect
1093 IF a=0 THEN RETurn
1100 NET 1
1110 OPEN #4, neto_2
1130 PRINT #4; CHR#(a);CHR#(13);
1140 CLOSE #4
1200 END DEFine help
```

QUALITAS

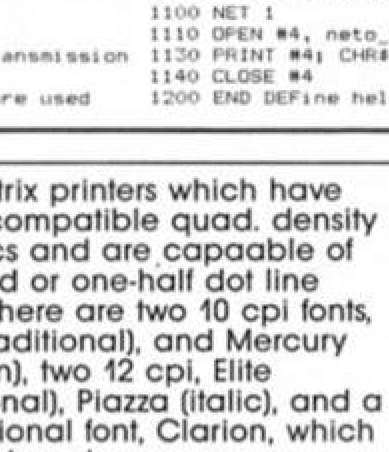
Qualitas
Seven Stars
Spectrum 48K/128K &
Tasword 2.
£7.95
(Tasword 3 version.
£8.95)

This is a handy utility for Tasword users, giving five Near letter Quality type fonts — all with equal space justification — on dot matrix printers which have Epson compatible quad. density graphics and are capaable of one-third or one-half dot line feeds. There are two 10 cpi fonts, Pica (traditional), and Mercury (modern), two 12 cpi, Elite (traditional), Piazza (italic), and a proportional font, Clarion, which gives a typset appearance.

The program, with your chosen main font, is easily merged with a version of Tasword alrady customized for your printer and interface. The 'help' pages are lost, text capacity reduced to 294 pages, and the only graphics printer controls available are underline and emphasized. To access the other fonts, you exit to BASIC and load the code from tape or microdrive. The program crashed instead of printing when a couple of the fonts were loaded using LOAD "" CODE, but all was well using LOAD "" CODE 50944.

Right: Screen dump of redefined character
Far right: Font editor showing letter with accent added





Print speed is slowed, because the head makes two passes across each line, but the quality is excellent. Even my rather tired old ribbon produced clear typscript worthy of a daisy-wheel, and the equal space justification gives the most professional results. If you need to align columns, character spaces are available which will not be proportionally justified.

There is also a font editor, enabling characters to be modified, perhaps by adding accents, or defining fractions instead of unwanted characters. For the really adventurous, a blank character set can be generated to allow you to design your own complete typeface. Screen dumps are available to help keep a record of your designs, or to produce paper copies of the blank grid on which to plan our characters. Your modified or new fonts can be saved to tape or microdrive.

Qualitas is a useful program, making professional quality printouts available to those of us who like the versatility of the dot matrix, but regret its sub-standard appearance.

Carol Brooksbank

ACROSS THE POND

U.S. Sinclair users are better equipped with disc systems than their U.K. counterparts. Mark Fendrick looks at what's available.

Back in the days of the ZX-80 (and later the ZX-81) the fact that we were able to save our programs and data onto a standard cassette recorder was nothing short of fantastic. True, they were slow and error prone, but they were inexpensive (most families owned one already so another purchase was unnecessary) and the cassette tapes were readily available. Since most of us bought these computers because of their low price, this suited us fine.

Often though, loading and saving programs was a major chore. First obstacle was the very slow cassette interface. To load or save a 16K program (data of course was included with the program and not separately) took them about 11 or 12 minutes. If you happened to have a 64K memory expansion, you would have to expect to wait as long as twenty-five minutes for your program to load. Sometimes after waiting for this to be accomplished, you would be greeted by some form of system crash, indicating a loading error. Making multiple saves was hardly the answer as the time involved was excessive. The success of a load was very dependent on the volume and tone set on your cassette player. Often a slight misadjustment would cause a loading error. In addition, it turned out that often, standard cassette tapes were not manufactured to exacting enough standards required for data storage. Computer grade cassettes were necessary for security of your data, and they could be costly.

Gone floppy

The rest of the home computer field had gone the route of floppy discs for mass storage. It wasn't long before a disc interface was available for the ZX-81. It was developed by the Acme Electric Robot Company (AERCO) of Austin Texas. AERCO was to become a name familiar to just about every American Sinclair user before too long. To this date, the AERCO name has more or less become a standard in Sinclair interfaces in the U.S.

ZX-81 owners had mostly bought that computer because of its low price, and for the most part did not spend the money required for the interface, disc drive, power supply and case. Not many ZX-81s were being used for business purposes (although I did use mine to keep my business records as well as for many other purposes) and did not justify this expenditure. When the T/S 2068 made its appearance, it was with an improved cassette interface — faster and less dependent on volume and tone settings. But it was the promise of microdrives that created the most excitement. They never appeared, but it was AERCO to the rescue once again with their interfaces for the T/S 2068. Most T/S 2068 owners would not be satisfied with using the T/S 2040 printer (and the ZX printer would not operate on the T/S 2068) and turned to AERCO Centronics and RS232C interfaces.

The system for the T/S 2068 is called the FD-68 Floppy Disc Interface. It is designed to control up to four Shugart compatible drives. These may be any size from three inch to eight inch, either single or double sided. Although eight inch drives are limited to single density, all other drives may be single, double or quad density. AERCO has drives available or you may supply your own. If you have drives from the T/S 1000 (or almost any Shugart compatible drives) they will work with the FD-68 interface. You may use any combination of drives you want since AERCO will custom burn the EPROM which contains the DOS for you. (AERCO does not charge to customize the original EPROM shipped with the interface).

The interface has an 8K RAM which holds the Disc Operating System (DOS) and requires no modification to the computer. RGB output is also available on the interface so that you may use an RGB color monitor. One of the exciting features of the T/S 2068 is the ability to bank switch additional RAM memory. The FD-68 interface includes an additional 64K of RAM, bringing your T/S 2068 up to a total of 112K RAM. This RAM is located in the Dock Bank and may be accessed in 8K chunks. The on-board RAM may be expanded to 256K if you require.

Once set up, your first step will be to run the System Boot Disc. The system will automatically seek out a program entitled BOOT. BAS on disc A and execute that program. With the System Boot Disc in place you will get the welcome screen containing the AERCO logo, and three menu selections; A — for an Amazing Demonstration which will show off the system; D — which will load the print driver(s) (both serial and parallel are included) and terminal emulator codes; I — for operating instructions. Pressing any other key will show the directory of the disc contents.

I.D.

The directory will show a disc I.D. number, title and memory remaining, followed by a listing of all files on that disc and the amount of memory remaining. You may have a total of 31 entries on any directory even though there may be space left on a disc. Any attempt to enter a 32nd file will be greeted with a "directory full" message. To access a program on the listing, it must be spelled exactly as it appears, although upper and lower case may be ignored. You can view the directory at any time by issuing the command, CAT "",. Viewing the directory does not disturb the system memory and you may return to what you were doing prior to calling up the directory.

The AERCO DOS makes use of many of the Sinclair keywords which were included in the T/S 2068 for future use, but which were never implemented. In order to use a blank disc, you will first have to format it by using the keyword FORMAT. FORMAT is accessed on the T/S 2068 by entering extended mode and then pressing symbol shift — zero. (This same procedure can be used to completely erase a disc as well). This will record track and sector headers on your disc. Depending on what has been programmed into the EPROM these can be the same or different for each drive connected. If there is data already recorded on the disc you are attempting to format, you will receive the following message: DISC A HAS DATA! OK TO ERASE? Y/N. If you choose any key other than "Y" the format will be cancelled. If you try to format a disc which has the write protect opening covered, you will be told: DISC A WRITE PROTECTED.

The CAT keyword (extended mode — symbol shift/9) serves double duty in this DOS. First it is used to list the directory of a disc; **CAT ""**, will show the directory of the last drive used. To catalog a different drive (such as drive "B" in this example) you enter **CAT "B:"**, and that drive becomes the default drive. If you replace the drive designation with the file name, it serves the same function as the LOAD keyword. A drive designation may also be placed in this command if accessing assorted drives is necessary. As in cassette loads and saves, a program which is saved as an auto-run program will execute upon loading.

The equivalent DOS command to save a program is MOVE (extended mode — symbol shift/6). A program can be made to auto-run by placing the line number after the comma; MOVE "name, ext",1. In this example, execution would start with line one of the program when loaded using the CAT command. Other parameters are necessary when saving other types of files, which we will examine in just a bit.

To remove a file from the directory, you use the Sinclair BASIC keyword, ERASE (extended mode — symbol shift/7). This will remove the file and increase the memory available showing on the directory. Files must be saved with a three character extension. The extension you use is dependent on the type of file you are saving.

.ARO is used to save or load memory into the Dock Bank, starting at address 8000H. This area is governed by the standard Timex rules (see the new Timex technical manual available from the Time Designs Magazine Company; 29722 Hult Road; Colton, OR 97017; U.S.A.; (503) (24-2658) for operating an AROS (Application ROM Orientated Software). The difference is that you will be running it in RAM instead of ROM.

.BAS is the standard save/load of a BASIC program along with its data. By adding the number of the line number to autostart from, you have the equivalent of SAVE progname LINE nn.

.BIN is used to save/load binary machine code data. As with the SAVE CODE command, you need to specify both a starting address and the length of the file in order to save the file. No parameters are required to reload in the same memory location, but you can reload to any other location by simply specifying the address.

.BUT would be used for an assembly language program written at 3410H (or D410H in Spectrum mode). It is the cold boot loader in CP/M and other non-BASIC environments.

.SCR would be used to save/load a screen. When the screen is saved, it will automatically include the second display file if it is active. Loading a file with the .SCR extension will return the image just as it appeared when saved.

.VAR is an extension which allows you to save/load all of the BASIC variables. It does not affect a program and can be used to manipulate the variables even without a program present.

There are three extensions which have not yet been implemented in the current version. They are **.CHR** (character data), **.DAT** (numeric data) and **.LRO** (runs in the dock bank starting at 0).

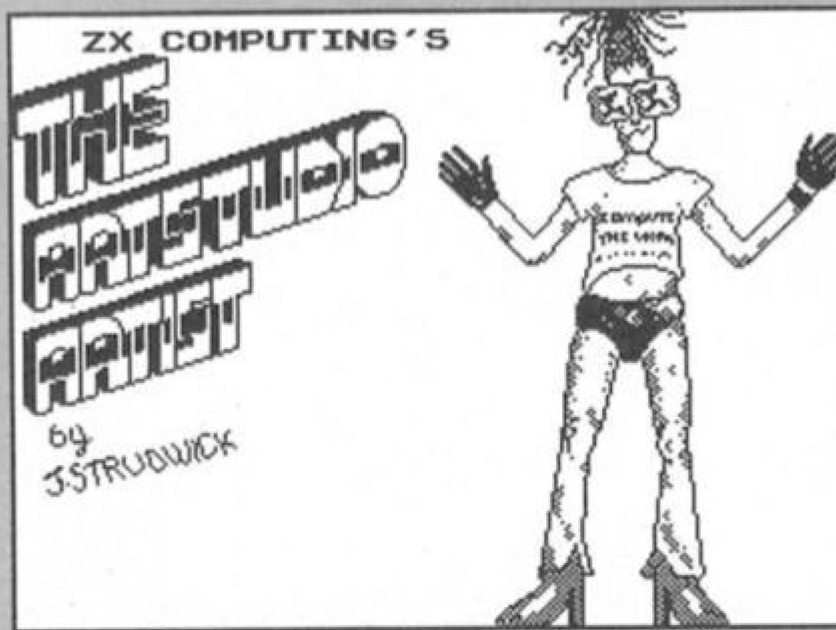
A program named BOOT.BAS will be autobooted by the FD-68 DOS. When powered up, the system searches for a program with that name, and if found will automatically load it. This program can either be a BASIC program, or a BASIC program which loads a machine code program. The AERCO FD-68 system comes with a very clear and complete 26 page instruction manual, complete with instructions, information about disc systems in general and many pages of diagrams and schematics. Even those highly critical of the state of documentation should be pleased with this guide.

Software

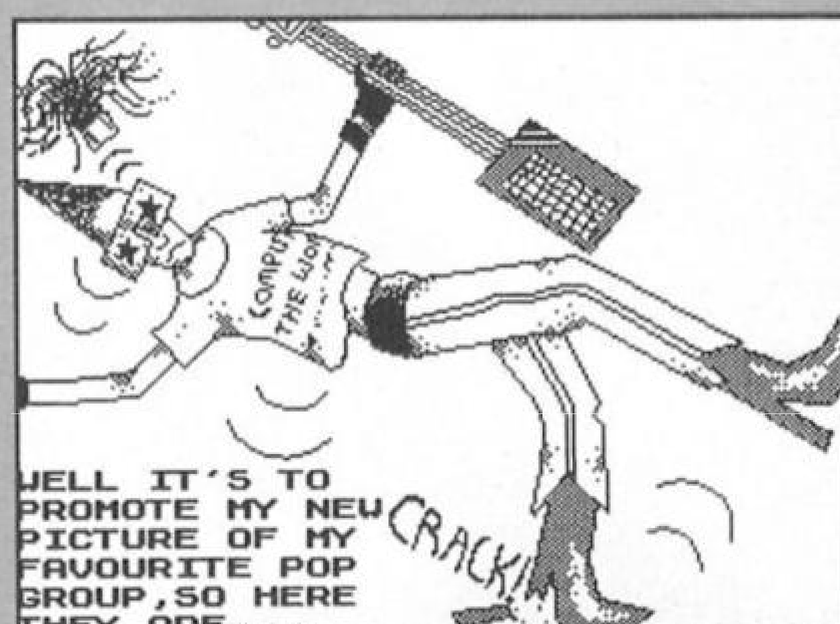
But the AERCO story does not stop there. One of the few regrets many T/S owners have is the lack of quality software in the numbers available for other machines. It is almost unimaginable for a U.S. Sinclair owner to walk into his neighborhood computer store and be able to buy software off the shelf. Not only that, but being able to select the "big name" software about which you have been reading in the computer magazines? A fantasy? Not any more thanks to AERCO. Not long after the T/S 2068, AERCO's ads started announcing the upcoming release of CP/M for the new computer. Before IBM made its appearance on the personal computer scene, CP/M (Control Program for Microcomputers) was fast on its way to becoming the industry standard. Many of the computers which appeared at that time had CP/M as their operating system. Many classic software packages, such as Wordstar, were released in the CP/M format and could run on any CP/M computer. A great deal of public domain software was, and is, available in this format.

But that was in 1984, and after a while, when the T/S CP/M did not appear it was all but forgotten. But not by the folks at AERCO. They were continuing with the necessary licensing negotiations as well as the actual development of the OS. Finally, not long before the Timex/Sinclair Computerfest in Cincinnati, RP/M (Resident Program Module) was released. It was given a showing at the Computerfest and was greeted with great enthusiasm and interest. RP/M is functionally identical to CP/M version 2.2 and software that will run under CP/M 2.2 will run under RP/M. Look for an in depth look at RP/M soon here in ZX Computing Monthly.

AERCO has been responsive to the Timex/Sinclair user since the day they first started supporting the line. Although their products are well designed and come with excellent documentation, there is always that stray question or problem that pops up from time to time. Help, in all forms, is only a telephone call away. I often hear complaints about lack of responsiveness from many companies claiming to support the end user, but only a few companies elicit the positive feedback I have always received regarding Jerry and Till Chamkis of AERCO. For more information write to AERCO (Box 18093; Austin, TX 78760; U.S.A.; (512) (451-5874). (British Spectrum owners can contact AERCO as well as they make their systems Spectrum compatible as well). Tell them you heard about it here.


ZX COMPUTING'S
THE ARTSTUDIO ARTIST
by J.STRUDWICK


YOUR PROBABLY WONDERING WHY I'M DRESSED LIKE THIS!


WELL IT'S TO PROMOTE MY NEW PICTURE OF MY FAVOURITE POP GROUP,SO HERE THEY ARE...
CRACK!


NEXT MONTH
loading problems

More tips for the faithful old '81 from Ray Elder.

First things first, a quick correction to last month's code. Yours truly must have been suffering from a surfeit of Spectrumese as I wrote the code. I put the NEWLINE character as 13, that of the Spectrum ENTER, instead of the ZX81's 118. Of course this means that it wouldn't work and, though most of you I'm sure put it to rights, some of our less experienced readers may be stuck. The remedy is simple. Replace *both* codes OD in the hex string (A$) with 76 and all should be OK. Sorry!

To business

This month I am giving some code to print out the characters entered by last month's input routine and then to develop an all purpose text storage routine that can be used with it.

A quick word for relative beginners. You will need a line 1 REM. . .with as many dots as you will have characters in all your text, PLUS 30 for the machine code and it is best not attempted unless you have a rudimentary grasp of machine code programming at least. So there!

But seriously, the code to print a stored set of characters is easy:

1. Load register pair HL with the start address where the text resides.
2. PEEK it into register A.
3. Compare it with 118, the newline character (!!!!!).
4. If it is 118 end and return from the routine.
5. Print it at the next character position.
6. Increment the HL address by one.
7. Go and do it all again by jumping to step 2.

And the code is given in FIG 1 ready for loading into your REM at whatever address you wish. Replace the ** with the hex address where you stored the code input from last month's routine.

As given it leaves the print position at the space immediately following the last character printed. If you move step 5 to between steps 2 and 3 then the print position will be placed at the beginning of the next line.

So far so good. It's easy to call it from BASIC by using PRINT AT y,x; first, but setting the x and y position from machine code needs a little program itself. One such is as follows, all it does is set register H to the x or column position and L to the y pos or line number and store it in the vars 16441.

In Mnemonics: **LD H,col : LD L,line : LD (16441),HL**

Or in HEX, **26002E15225764**

And then call FIG 1

Text storage

While this is fine for printing text entered from our input routine it can also be used to print longer previously stored text for instructions, messages, adventure games etc. and so I offer FIG 2 as a means of storing larger text easily.

To use it decide where in your REM you wish to store it and enter that address at the start. Several sets of text can be stored immediately after each other and the start addresses are given at each stage. Make careful note of these addresses for later use in loading the HL registers when you wish to use them.

There are many variations which we could use, for instance by including the code 21000D097EFE76C8D72318F8 at the start of each text item all that would be required to print it would be a simple CALL to its address, but this is merely a suggestion for advanced users to play with.

And so. . .

Once again we come to the end of our ZX81 section. I would like to make my customary plea for routines or even short ZX81 programs or suggestions for further developments to our REM which you would like.

Bye. . .

FIG 1.

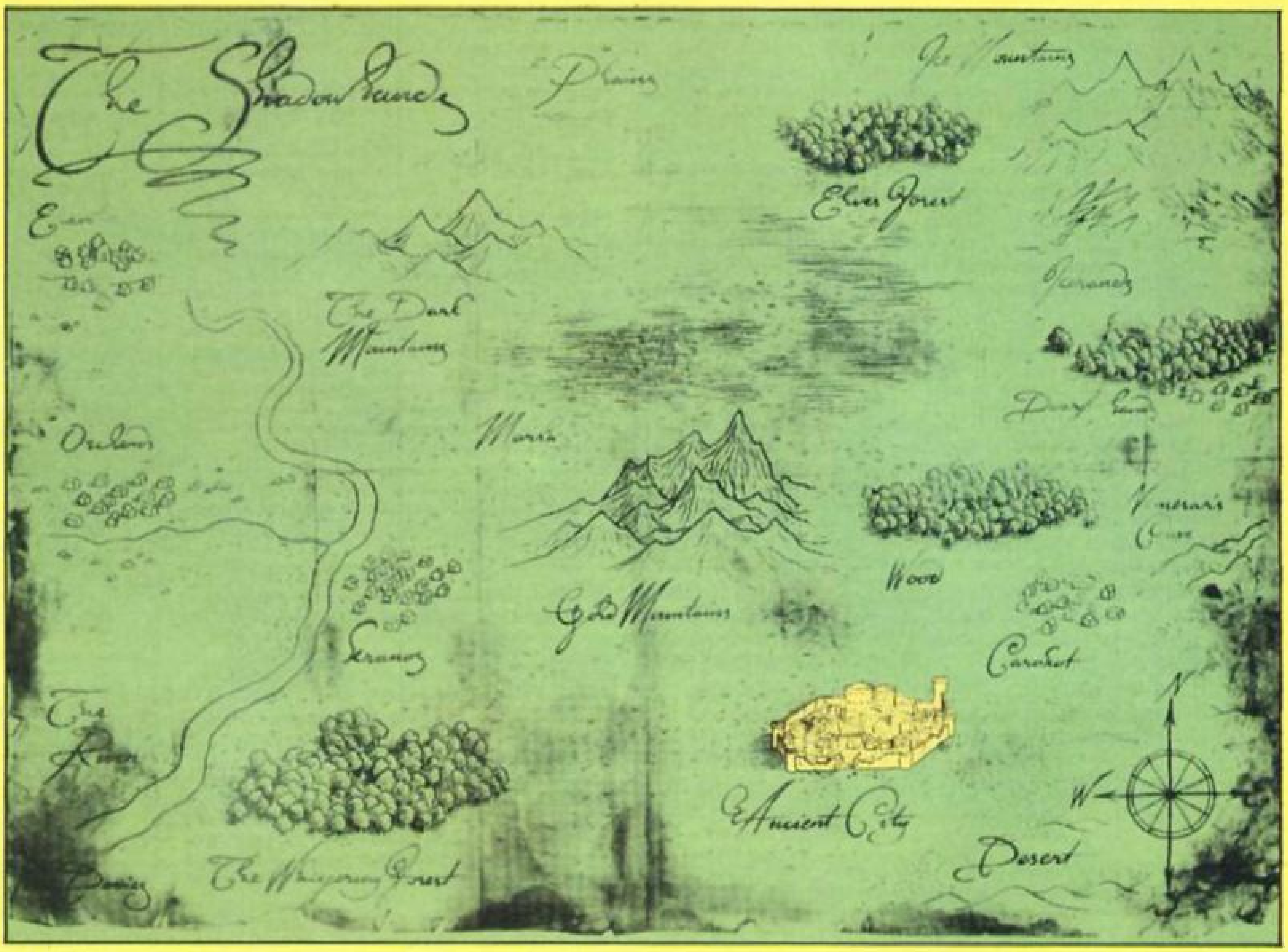
```
   1 REM ...ALL YOUR
DOTS........
  10 LET
A$="21**7EFE76C8D72318F8
"
 100 FOR I=17500 TO 17510
 101 REM SET ABOVE ADDRESS TO
SUIT YOURSELF
 110 POKE I,16*CODE A$+CODE A$(2
)-476
 120 LET A$=A$(3 TO )
 130 NEXT I
```

FIG 2.

```
  10 DIM S$(500)
  20 PRINT AT 21,0;"ADDRESS TO S
TORE TEXT"
  30 INPUT A
  40 LET N=1
  50 CLS
  60 PRINT AT 0,10;"TEXT ENTRY P
ROGRAM"
  70 PRINT AT 21,0;"ENTER TEXT I
TEM ";N
  80 INPUT T$
  90 PRINT AT 2,0;T$
 100 PRINT AT 20,0;"AT ADDRESS "
;A
 110 PRINT AT 21,0;"IS THIS CORR
ECT Y or N)"
 120 INPUT A$
 130 IF A$<>"Y" THEN GOTO 50
 140 FOR I=A TO A+LEN A$
 150 POKE I,CODE T$(I+1-A)
 160 NEXT I
 170 POKE I,118
 180 LET A=I+1
 190 PRINT AT 21,0;"MORE TEXT EN
TRIES  "
 200 INPUT A$
 210 IF A$="Y" THEN GOTO 50
 220 STOP
```

PAGE 81





A familiar adventure tale given the icon driver treatment by the Edge.

Mindstone
The Edge
£8.95

It's the same old story, handsome young prince born first in line to the throne and doted over by his parents whilst the scraggy younger brother gets all jealous and twisted and runs off with the magical heirloom that gives the king his powers and keeps everything jolly in the kingdom.

That's the story behind the new icon driven, animated graphic adventure from The Edge, called Mindstone. The Mindstone is the magical object that has been stolen by the evil prince Nemesar, and it is your task in the game to take on the role of Kyle (the crown prince) and to lead him and his followers across the Shadowlands to confront Nemesar and regain the Mindstone.

Kyle's three followers are Merel, Taina, and Quin, and like Kyle they all have their own Strength and Psyche scores which determine how succesful they are likely to be in physical or magical combat. Each of the four characters can carry four objects and a single spell, so there is an element of judgement involved in deciding how to allocate the items that you find on your journey.

Merel is physically strong but psychically weak, whilst Taina is the reverse. Quin and Kyle are about average in both areas, so obviously you use Merel and Taina as specialists in their respective areas while Kyle and Quin are really just back-up men as far as combat is concerned (though when their strength scores are at their peak they are both capable of knocking out the odd wolf or crocodile. And knowing just when to use which character is part of the skill of the game, Merel and Taina might be the specialists of the team, but if you use them too often you'll drain their abilities and find yourself lumbered just when you need them most.)

It's a bit tricky to describe Mindstone as it incorporates elements from games as diverse as Shadowfire, Tir Na Nog, and even a bit of Lords of Midnight, but basically the game is a graphic adventure with some animation, icons, and a few other bells and whistles added on. That's not intended to sound too derogatory, since the whole thing is very well presented and quite absorbing and doesn't make any claims to be anything other than a graphic adventure, but perhaps I was expecting something as innovative as Fairlight, The Edge's last trip into arcade adventuring.

The top third of the screen is used to show a graphical representation of the current location, along with any other characters, gateways etc that

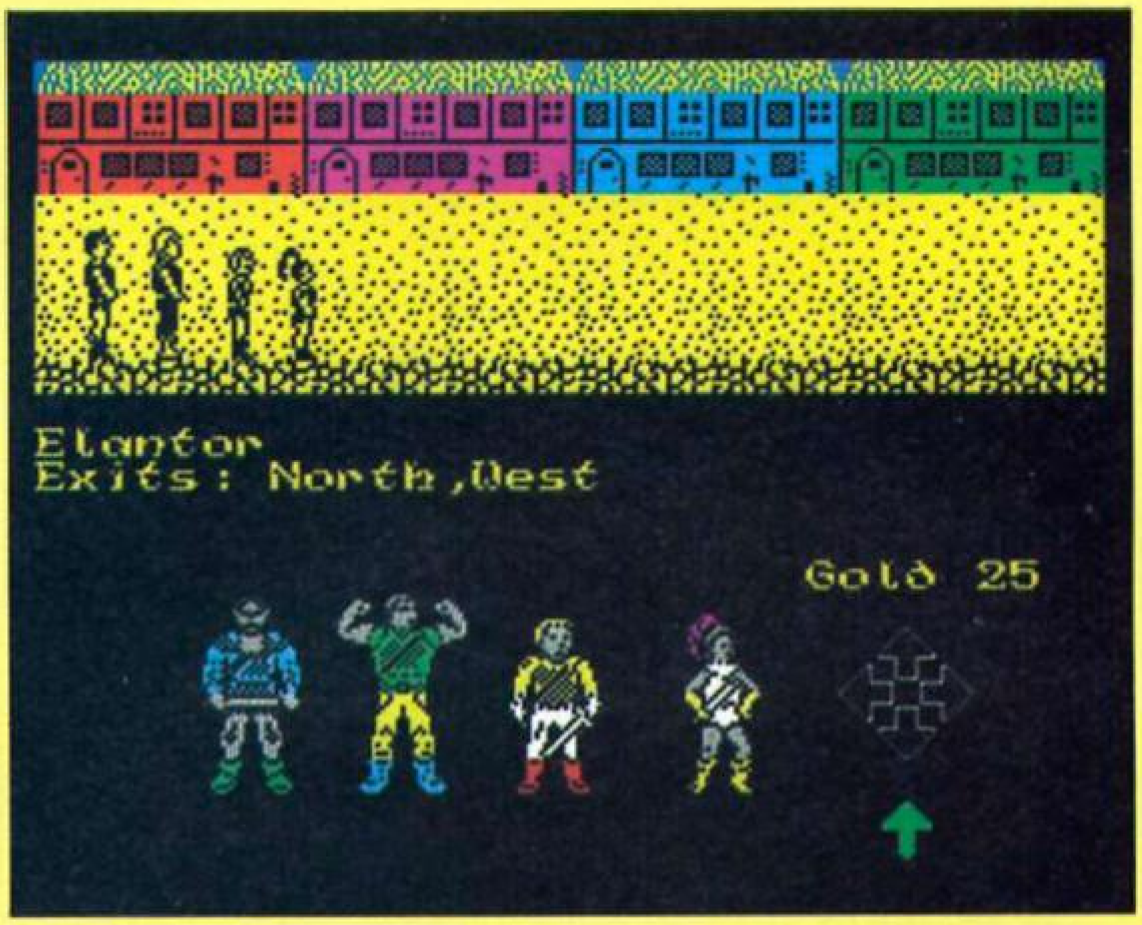
you might have to deal with. Your four characters are generally shown standing at the far left of this picture, but they can be seen to walk across the screen in response to your commands as they attempt to attack or trade in some way with other characters.

The remainder of the screen display contains icons of the four characters and a compass which you use to direct their movements. The author, John Pitcher, has allowed for both icon control of the game or a more conventional keyboard control.

If you choose icon control you can steer a pointer around the screen using a joystick, and once you have chosen the character you wish to instruct you can choose either an Actions or Attack menu which offer you a range of options for combat, spell casting or for dealing with objects or other characters. These include obvious adventure commands such as Attack, Examine, and so on, but also a number of others including Trade, Buy, Meditate, and Dig. In icon mode you choose the appropriate command by moving the flashing cursor around the menu, but in keyboard mode you have a system of single key entry (as in Lords of Midnight) whereby each key corresponds to a specific command. Keyboard control means that you've got more to remember, but icon control is a little bit fiddly at times as you keep swapping from character to character and all the various menus. But the control system does work quite well, as the icon system allows you to get started quickly and without having to memorise all the control keys, then, as you become more accustomed to the game, you can speed up play by moving onto keyboard control (but you can only use one control system in each game, you can't swap from icons to keyboard in the middle of one).

As with all adventures it's vital that you make a map as you go along, although The Edge thoughtfully provide a rough sketch map of the Shadowlands on the cassette inlay. I suppose that because of the animation I was expecting more of an arcade element in the game, perhaps something like Enigma Force or Fairlight, but as I mentioned earlier Mindstone is really just a graphic adventure with souped-up icon driven presentation, and once you treat it as that both you and the game are more likely to do each other justice.

As with most adventures, problem solving is a matter of gathering together the correct objects in the right place,





though, as usual, there are always more objects to carry than you can use at any one time so you've got to choose which objects to take with you and which ones to leave behind for later use. When bartering for items or deciding which character to use in a fight there's the added element of judgement that you've got to use and quite often the consequences of an action won't become clear until much later in the game.

The memory taken up by the icon system means that Mindstone isn't as complex or as rich in brain-bubbling puzzles as a Level 9 game might be, but on the other hand the control system is very 'user friendly' and allows you to get into the game quickly and without having to sit there wondering which is the right word to use all the time, so Mindstone is well worth having a look at whether you're an adventure novice or a hardened orc basher.





MINDPLAY

Peter Sweasey reflects on the torrent of Tolkien based adventures and gives a selection of helpful hints to budding adventure writers

"I have no use for adventures. Nasty disturbing uncomfortable things! Make you late for dinner! I can't think what anybody sees in them." That immortal insult was uttered by Bilbo Baggins in J.R.R. Tolkien's The Hobbit. July was Tolkein month, according to Melbourne House. Being something of a cynic, when this was announced earlier in the year, I dismissed the idea as a particularly crass promotion designed to sell yet more copies of an old game.

In fact, July did turn out to be Tolkien month, to my great surprise, and probably to the even greater surprise of the PR people. Because before the month was over, I received three computer games based on the exploits of creatures with leathery-soled hairy feet who are half our height, who live in holes and encounter trolls, spiders, wizards and dragons in a reluctant quest for treasure. All three are extremely good too. I was never a great fan of the Tolkien — I enjoyed The Hobbit when young but found the Rings trilogy dull — so it's a pleasant surprise to find three superior games. Bilbo would have been horrified to thing he was responsible for such madness.

We also have some budget titles, including a rerelease of The Hulk from US Gold. I haven't given it a good review because it's basically a feeble adventure, but I hope to see other Scott Adams games at this price. Spiderman, Claymourge Castle and other of his better titles would be perfectly suited to a £2.99 tag.

Finally, before the reviews start, a second look at Hunchback The Adventure. In my review last month, I said I would reserve full judgement until I had at least reached part two. Now, thanks to John Wilson, I have completed the game, and my verdict was quite correct! The solution is as I suspected mostly finding objects and dropping them in the right places. It is

likely to appeal most to inexperienced adventurers, and people who don't play often. Although on a general scale my rating stays the same, if you are an occasional adventurer it is raised to "Great".

Dungeon do's and don'ts

What does an adventure reviewer actually do? Unlike arcade games, we cannot comment on animation, sound and basic categories like that — so is our job just opinion? No way! I take into account a wide range of points. I thought any readers interested in writing a game — and judging from my post, there's an increasing number of "amateur" authors — might be interested in hearing some thoughts from a reviewer. After all, if you want to sell a game, either myself or a colleague in the reviewing world will have to see it sooner or later, for it to gain the necessary publicity and (you hope) acclaim). So why not cut out what I will criticise before you send it to me?

The first point, and one of the most important, is plot. IF YOU CAN'T THINK OF AN ORIGINAL PLOT, DON'T WRITE AN ADVENTURE. Except maybe if you are a really stunning writer, and even then my reception could be cool. No plot is totally original of course; but some of your game should be reasonably new. By plot, I mean scenario above all. Please don't set it in outer space or a cave network... I've seen it all before and believe me, so have many adventurers. However, if you follow all the other rules I outline, you might just get away with it.

Make the plot logical. Don't do what one writer did recently, and have a genius leave parts of a secret code scrawled all over the floor. Objects should be in logical places, unless your game is humourous. That area is one of the hardest to succeed in — humour is many things to many people — and should not be seen as an excuse for plot short cuts.

Text should be atmospheric and lively. Long text isn't necessarily good text — it can easily be wasted — but location description should be at least two or three lines long. Scott Adams style is too brief: never use his games as a standard as they're very poor examples of adventure writing. As for graphics; unfortunately if you want commercial success, they're almost essential. If you put some in, take some time on them, and make them split screen. The full screen "Illustrator" graphics are slow to draw, obtrusive and unprofessional; so use GAC or Patch (more on which later). Quality is more important than quantity — a few excellent pics are better than many poor ones.

Vocabulary should be as friendly as possible. Use plenty of synomns. Get a friend to playtest your game; often he will think of word combinations you hadn't. The adventure should not be testing the player's ability to guess what the author was thinking. EXAMINE is almost vital these days. SEARCH is useful. LOOK and the abbreviation L should redescribe a location. If you have other characters in your game, a TALK or SAY command greatly enhances play. If an object is mentioned in the text, it should be understood unless it is clearly just description.

The game should allow the player to do things that are not necessary for the completion of the game. Let him go up a complete blind alley if he wants, with messages to encourage him. It adds to the fun and the sense that the computer is doing what the player wants, not vice verca. For example, allow him to EXAMINE everything. This means many messages, but is worth it. As a general rule, the response "You can't" should appear rarely. A reason why not, or what irrelevant action happens if you do, should instead be used. Again, thorough play testing by friends causes them to try actions which you hadn't though of, and can program a response to.

If the utility you are using runs out of room for messages, locations, flags or whatever — split the game into more than one part. If you use the Quill, nowadays you must use Patch with it: this allows split-screen graphics and the ver useful RAMLOAD and RAMSAVE commands, which greatly add to a game's playability. Make sure you use series C Quill — it's less memory consuming and more flexible.

Presentation is extremely important. Use colour, but nothing garish. If you are using a utility, redefine the system messages otherwise the game looks as if it has come off a production line. Define a more readable or suitable character set. Take care with spelling and particularly punctuation (why do so many writers know so little about apostrophes — make sure you know the difference between it's and its).

Play other games. Level 9 give you something to aim at! If you're using the Quill buy The Boggit by CRL to see how to use it properly — a near perfect "amateur" game. In my experience keen adventurers are often the best games writers: they know what they dislike and like about other games and aim to improve on them. If you couldn't care less about adventures, and just want to write a money-spinning game, give up now.



Once your game is written, marketing it yourself through classified ads is unlikely to bring much response. Having a software house publish it is a better solution. Very few Quilled games are sold for full price, but the budget market is thriving. From what I've seen of games by Atlantis and Central Solutions, their standard is not too high, so there's plenty of room for your improvement.

The emphasis in this article has been on commercial success: the reason for this is that, after all, my job is to recommend to the public which game to buy. Finally, the best of luck with your adventure-writing career — I hope to give your game a Monster Hit soon! Every adventure sent to me is looked at, but these days only good products from small companies gain printed reviews.

Footnotes: "The Patch" is available for £5.99 from Gilsoft, 2 Park Crescent, South Glam. The use of "him" for "the player" in this article is purely for convenience and does not represent any prejudice against women using computers!

HELPLINE

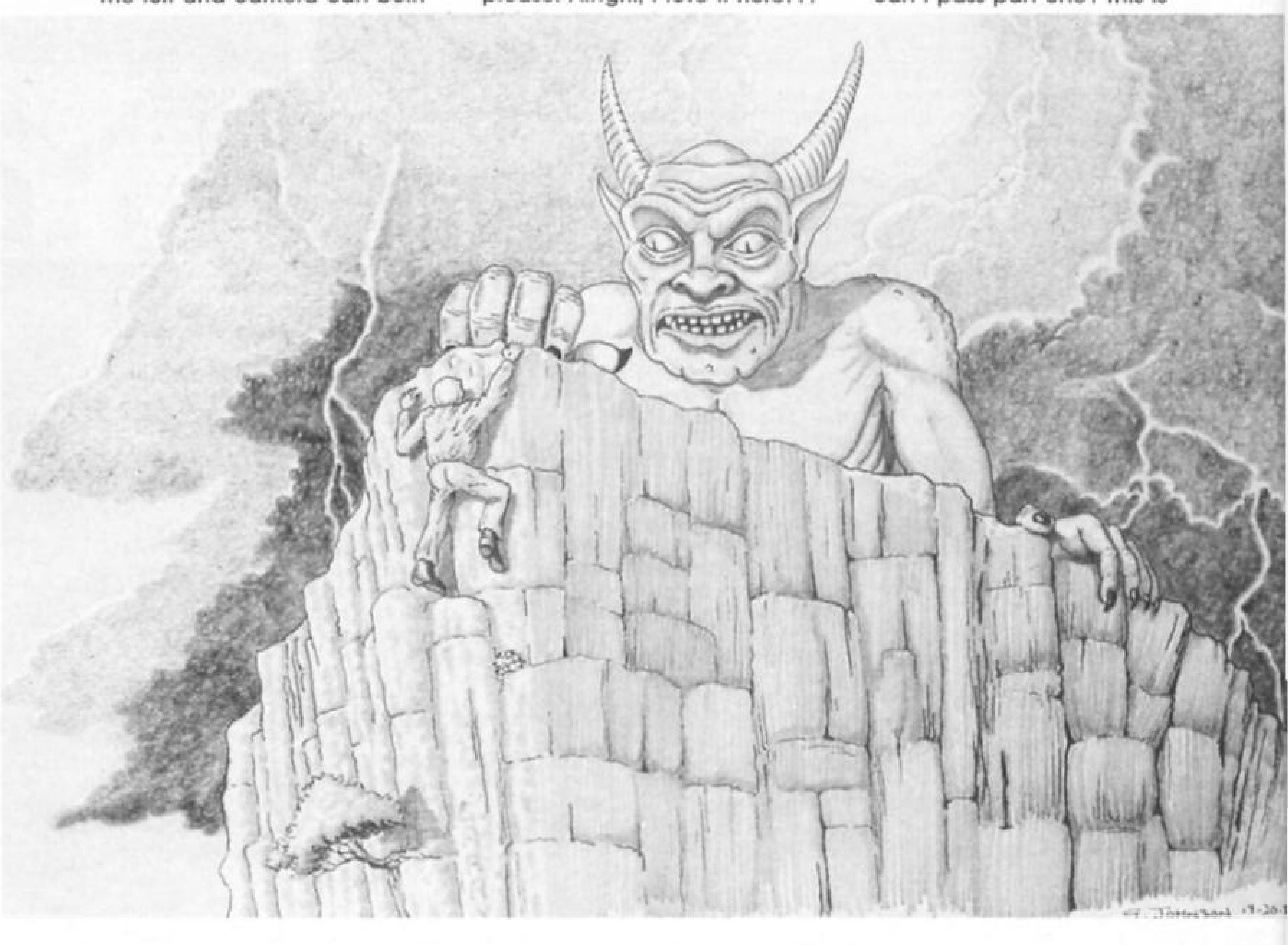
After last month's massive missive on your mail, back to a more manageable length. Where better to start than with *The Hobbit*? (coming out of my ears by now). Niclas Carlenius from Norrkoping in Sweden is stuck in the Goblin's Dungeon, like so many other brave adventurers before him. To open the trapdoor, you must BREAK it several times. To escape, you must wait for Thorin to appear, then SAY TO THORN "BREAK WINDOW", tell him to "CARRY ME", then to "GO WEST". He also asks how to reach "'OUTSIDE Goblin's gate" when he's found the ring. You need to WEAR it, and as it keeps falling off (though the program doesn't immediately tell you), it's best to try this command before every move. Go N, SE, W, N, D, S, W, E, and OPEN DOOR.

Jon Wilde wants to go to *Terrormolinos*, from Wombwell in Yorkshire. But he asks "How to pack all the clothes and equipment, how to get the camera and to reach the loft". The loft and camera can both be reached with the steps from the garden, but LOCK them before use. Then CLIMB or DESCEND STEPS. As for packing, just take everything you feel is necessary, plus the suitcase, when you board the taxi, and the program will do the rest. You cannot take the shaving foam and harpoon gun together; you only need the latter. Thanks to Paul and Glenn Gibney for sending a complete solution to this game, which I've only recently obtained. Old correspondents from my days at Home Computing Weekly, they write "It's great to hear that you've been replaced in a new damp, dark dungeon with all mod cons". Ha! This is unfortunately the same grotty cell I've always been in (Argus only has one), and unless you call the two year old straw which lines the stone floor a "mod con", I'm afraid you're mistaken on the last point. A pitiful shame, it (stop moaning or I'll send Leslie Crowther down — Ed). No, anything but that, please! Alright, I love it here...

P. Ellaway wites about the superb *Fourth Protocol*. He's scored 98% in "The Nato Documents" but cannot progress to the next part without the other 2%. The question would be impossible to answer if he hadn't sent me a complete list of what he'd done. Turns out he was giving the wrong answer to some of Plumb's questions, which are admittedly somewhat ambiguous. These are the responses you need:

Who is the traitor? — FAULKNER
What nationality is his contact? — SWEDISH
What name is he using?—NILSON
What is his real name? — STENBERIJ
What proof do you have? — FINGERPRINTS
Where did it happen? — GAZA
How was the traitor recruited? — FALSE FLAG

Then provided you have done everything else, advance the date a few days and a message will appear. Burak Karsu writes from Istanbul — yes, the one in Turkey — with the question "How can I pass part one". This is



rather vague as I don't know what you have done, but here are some general hints. When Bracton calls about Thorn, read Bracton's file, then put some watchers on. When you learn about his peculiar habits, 'phone medical security. Just analyse the leaked papers, soon Willis will be caught and you can prosecute him. Keep a large watch on Warburton, Banister and JOhnston when you receive information on them; also Faulkner, access PIZZA file when you're told where he eats. And in answer to Jason Gibson's question, to find the telephone numbers, ask for TELEPHONE file.

A couple of people have asked for help with Artic's ancient *Planet Of Death*. I'm not sure why anyone bothers playing it when there is so much better stuff around. Still, Douglas Harter from Enola in America keeps being killed by the green man. You need to TAKE MAN, DROP MAN (this moves him from the mirror) then SHOOT MAN with the laser gun. In case you haven't found that, take the floorboard from the house then USE BOARD to cross ravine. Take and wear the gloves from near the man. To pass the force field, FIRE GUN — AT FIELD twice, then DANCE. That should answer the queeries of Peter Davies of Lancaster, who also asks how to escape the prison cell: just avoid it, you don't need to visit it to complete the game.

Gerold Hofheinz writes from Stuttgart in Germany (a real jetsetter of a helpline this), asking for the five answers to the guardians before Vran's Inner Temple. John Wilson supplies the answers. Guardian 1: SAY 0,2,6,9. Ignore the second guardian. Guardian 3: SAY ANAGRAM. Guardian 4: SAY G A. Guardian 5: SAY TLE.

Andrew Jone addresses me as "Sir/Madam" — at least that's better than what Bryan calls me — and cannot ski in *Valkyrie 17*. He has the skis; well you need to wear them , and to take the ski poles from outside the hut (they're illustrated but not mentioned), and to have read the book. This is found south from the hotel room where you found the sheets. When you reach the bottom of the slope, there will still be some blood, though fortunately not yours.

Brian Coulthard asks some questions about *Runestone*. These are the answers. Questor's lake is found in the Deserted Lands (see the map) of the north, and is useful for rejuvenation after battles. To enter Kordomir's domain, follow these directions. Unless otherwise stated, you should move in the same way until blocked by mountains. From Saromundor's hut, N, W, S, W, S, then W unti lyou are able to move north, N, W until you are able to move north, N, E, N, E, N, W, S, W, N, In. The purpose of the Runestone will become clear at the end. But to destroy Kordomir, you need to decode the messages on the manuscript and ring; the latter object is vital for success.

Write to me

Whether you're baffled by The Boggit, mystified by Mindshadow, confounded by Colossal or stumped by Spiderman, we can help. Fill in the coupon and send it to: Mindplay Helpline, Deepest Dampest Darkest Dungeons, ZX Computing, No. 1 Golden Square, London W1R 3AB.

A few ground rules: if you're writing from Britain, please enclose a stamped, addressed envelope for my personal reply. If you are writing from abroad, please lay out your address in a way that I can copy on to an envelope, and will reach you. I try to respond within two months but I can take longer (on the other hand, you might receive an immediate reply). I'm not too hot on arcventures these days (Gargoyle games included). Finally, please write the name of the game(s) concerned on the back of the envelope.

Next issue the new Level 9 blockbuster will almost certainly have reached me. And it's the PCW show soon — before I next write, though after you read this — which could mean lots of lovely new adventures to review. Then again, considering last year's show, it could mean nothing of the sort. Find out in the next enthralling episode of . . . Mindplay!

MINDPLAY

JOURNEY TO THE CENTRE OF EDDIE SMITH'S HEAD

Players
£1.99

Eddie bumped his head on a nuclear reactor, has gone bonkers and is threatening to blow the world up with a bomb hes developed. You must stop him by investigating the case and defusing the device. Unfortunately the rather wonderful title seems irrelevant; you don't even get near his earlobe. At least not in what I've seen, and I haven't investigated more fully because this is not hot.

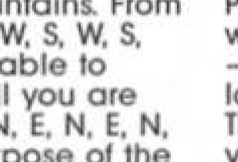
I'll show you why. The first location:

"You are in your own laboratory. Shelves are stacked with bottles and test tubes". (One of the more detailed descriptions; usually they're only one sentence).

After the possible exits, "I can also see:— A Flask, A Coat"
EXAMINE FLASK: "There's nowt special about this" (Is it empty or full? What is in it? There is "nowt special" about virtually everything you encounter).
GET BOTTLE: "It's not here" (Yet it's in the description)
GET TEST TUBE: "I can't (This turns up after so many commands).

Then there's the gun which you cannot shoot people with and the coffee which cannot be drunk. You die of dehydration despite passing through rooms with taps in them: they cannot be used. And so on: the author has allowed for nothing except the commands needed to solve the game; and those commands are hard to find because the vocabulary is fairly poor. There are some, full screen graphics; some effective, others dull.

The whole thing feels so tired; a waste of energy for all concerned. Not a stunning debut for a new label.



GRIM



AN EVERYDAY TALE OF A SEEKER OF GOLD

John Wilson
£1.99

The author of this game is John Wilson, an extremely accomplished adventurer who supplies many of the Helpline solutions. He's used all his experience to create an excellent game. John is a great Fergus McNeil fan, and became so fed up waiting for The Boggit that he wrote his own version. This isn't to say this is a rip-off of that game (it reached me some time before CRL's was released); it is much less derivative of the original as well.

AETOASOG if text-only and Quilled, but nicely presented. The text is quite lengthy, and like in McNeil games is not really description but humorous statements. In John's game the humour is much more gentle, with a clever idea or sentence causing a smile, as opposed to the complex set-up of jokes in The Boggit. This makes a pleasant change.

You face numerous ▶ ▶ ▶

problems on your way; several confront you immediately, and you are unsure which to try to tackle first, in the hope that it will enable you to solve the others. Mostly, the puzzles are fun, though requiring much thought. Vocabulary is, with some exceptions, friendly; and the game is pleasingly responsive to a range of commands.

AETOASOG is not a classic adventure, but is the best I've seen at the price — for two pounds, exceptionally good value. Virtually a must for all but novice adventurers.

Contact, John Wilson, 23 Spotland Tops, Cutgate, Rochdale, Lancashire OL12 7NX.



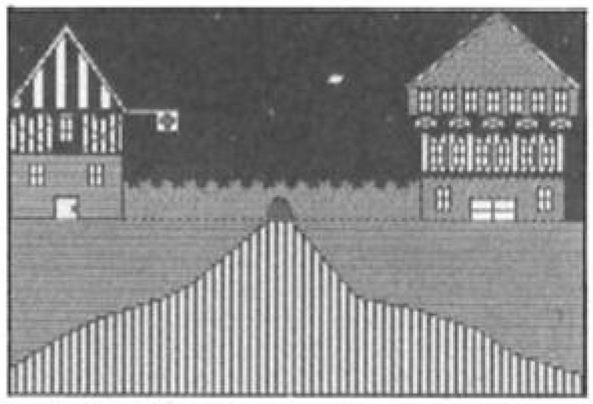
CLAWS OF DESPAIR

Players
£1.99

More a cause of despair as it turns out: another less than wonderful offering from Interceptor's new Players label. It is somewhat better than JTTCOESH, but still suffers from similar flaws.

Your quest is to recover a Staff of Saffrin, which will save your land for reasons not quite made clear in the slightly incoherent instructions. The game — what I've seen of it — is set in a middle earth type civilisation, with inns and a walled city, plus a wizard and other magic.

Description is above average, generally well written. The full screen graphics are mostly poor though, and excruciatingly slow. This is particularly noticeable, and



frustrating, in a desert maze, where the same rather dreadful picture is drawn at each of the many junctions; and since you are killed frequently, you have to reload often, when the pictures are drawn again.

Indeed, death lurks round unfairly virtually every corner, and usually there's no turning back, making play frustrating, not challenging. EXAMINE is rarely responsive, and there's an unnecessary pause each time you use it. The game is generally unfriendly. The problems are tricky, but quite original and fun.

The packaging is well designed but otherwise I'm totally unimpressed by Players. A much better idea would be to re-release some old Interceptor games like Heroes Of Karn: they're better adventures, ideally suited to a budget price and have delectable graphics. Players is not budget adventuring at its best: buy Seabase Delta or AETOASOG instead.



THE INCREDIBLE HULK

Americana
£2.99

A re-release of the first Questprobe game, courtesy of US Gold's budget arm. You play Bruce Banner and his monstrous alter ego, and, in the author's ever-realistic style, must collect some gems.

This was the first Scott Adams game I played and I was bitterly disappointed. The major fault is an intrinsically poor design. It relies on dirty tricks like having three different domes but identically external descriptions of each, and random distances between them (an astoundingly dum idea). Nearly half the gems can be found by entering three simple commands in various places.

The usual Adams flaws are here, like dodgy vocabulary and very sparse description (complete with "TM" after some words). Add to this poor quality programming (thankfully it improved with later games). Among the ways this manifests itself are the sloppy presentation of messages, and a peculiar parser bug which means some commands must be entered twice before they're understood.

The graphics were stunning when this first came out; now some look scrappy, but most are still rather good. Shame they black out every time you enter a command.

There are a few good points. Sometimes when you type EXAMINE you are shown a close-up picture, a clever idea; and the pics illustrate actions as well as locations. The instructions are impressively lengthy for a cheapie. And a few of the puzzles are fun, if you can tolerate all the faults which make them harder.

Perhaps worth looking at if you're an Adams or a graphics fan, and certainly better at £3 than full price . . . but still pretty naff.





THE HOBBIT

Melbourne House
£9.95

When The Hobbit first came out, I was not as amazed as all the reviewers of the time, in fact I disliked the game. I was unimpressed by the "independent" characters (all that "Thorin sits down and sings about gold" rubbish); and I objected to some of the hefty £14.95 (which seemed vastly expensive when compared with an average of £5 per cassette at the time) paying for a book which I, like many other people, already owned. Since then I have become an adventure reviewer and know what makes a good game. The ridiculous hype has died down and I can appreciate the program's good points, which are numerous.

Furthermore, I can review it honestly without being in awe. That said, it is deservedly a classic — and the game responsible for introducing so many people to the wonderful world of adventures.

The program has aged remarkably well; its speed and presentation still compare favourable with anything on the market, as does the friendly vocabulary and the responsive, helpful replies. the graphics are variable; some duff, some still good, but none really poor. They tend to improve the further you progress.

The location descriptions are far briefer than I remembered, perhaps too short by *today's standards*. However, with all the messages and characters wandering around, there remains a reasonable amount to be read.

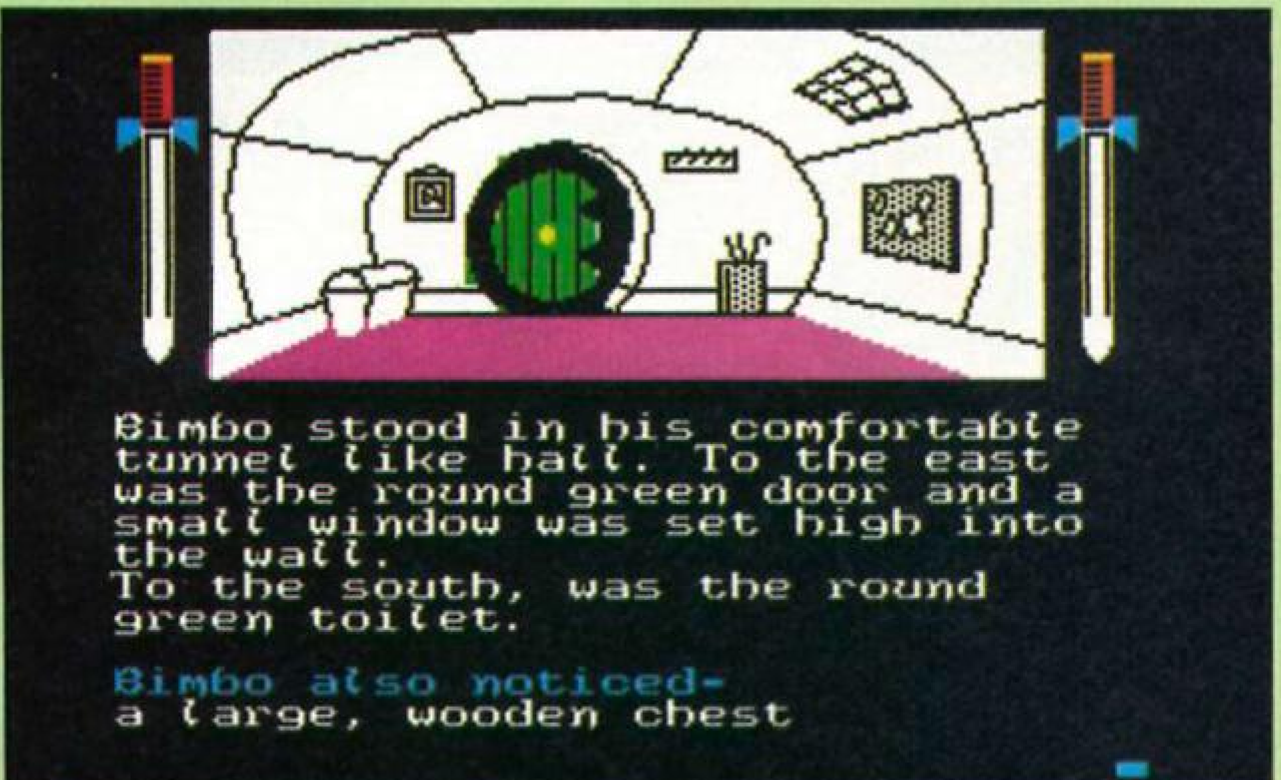
The game has some lovely — and now classic — puzzles. You haven't lived until you've escaped the Goblin's dungeon! Another admirabale feature that is rare even today is that many problems can be solved in a variety of ways, as can the game itself; gaining the treasure is by no means the sole task. This helps add realism to the computer controlled world and I begin to see what everyone was raving about. Character interaction is a vital part of the game, and well handled, though not so mind-boggling these days. What did happen, by the way, to the other twelve dwarves?

In many respects I prefer this to the first Lord Of The Rings game. The Hobbit has nothing like the same complexity, but is much more enjoyable to play, and I feel captures the atmosphere of the book better (no photographs or heavy metal here!). Then again, The Hobbit is a far less complicated story.

The only real problem is the price. I would have thought past sales would have paid for The Hobbit's development many times over by now; yet at £10 there's virtually nothing more expensive. Considering the book is no longer included, its price has hardly changed. I'm sure some shops have been selling The Hobbit off as old stock for much less, and at a mid-budget price of £5, or as one of Elite's "Classics", this would be a monster hit. As it stands, I would say it's still worth playing, but make your own judgment as to whether it is the best use for your adventuring budget.



THE BOGGIT





THE BOGGIT

C.R.L.
£7.95

No matter what the publicity says, The Boggit is a direct spoof of Melbourne House's game as much as Tolkien's book. Or is it a rip-off? It's a thin line between the two which author Fergus McNeil doesn't always walk successfully, certain problems being identical. That said, The Boggit is one of the best adventures I've reviewed this year.

The sheer daftness of the humour is what appealed to many people about McNeil's previous mainstream releases like Bored Of The Rings, and devotees will be pleased to know the same zaniness occurs here. At the start Bimbo Faggins has secured himself in his hole, determined to avoid any of those increasingly popular adventures. Before long, meddling old Grandalf crashes through a window, deposits a message with a self destruction box of chocolates and swiftly exits, all to the tune of the Milk Tray ads. When you eventually leave your home you are battered by Thorny and a bunch of dwarves into participating in Grandalf's peculiar version of "The Price is Right".

Soon the dwarves sing a charming song about how they're taking you (who they call "lardball" and "cement head") along on their mission (which I hardly need detail) so the dragon will eat you instead of them. Later you'll encounter the highly dubious Smelrond, the River Anadin ("great torrents of soluble pain relief roared quietly by"), and my personal favourites, the spiders who bob around "most convincingly" on elastic bands. Perhaps unsurprisingly, lavatories also feature strongly. The humour isn't to everyone's taste — I didn't find it hilarious — but there should be something to make most people smile.

Vocabulary is mostly friendly, with the exception of the word CLIMB, which is often used instead of ENTER, e.g., CLIMB INTO CHEST. The ever useful RAM SAVE/LOAD option is included.

The split screen graphics are all extremely attractive; with the exception of Adventure International's, I'd say they're the best I've seen. Many of them are close copies of Melbourne House's, and somehow manage to be both uncannily accurate and an improvement on the original.

There's more text than in the original too, with quite long, usually witty, location descriptions, verbose messages. Some events are described by several pages of text. The game is not always as responsive as I would have liked, but EXAMINE is usually productive and you're allowed to do plenty of irrelevant actions. Another nice feature is the characters who wander round in convincingly independent style; there's a clever TALK routine too. And yes, Thorny does sit down and sing about gold — though his song is somewhat cruder in this game.

There's lots to see but not that much to do in the way of actual problems. There's one real stumper close to the start, otherwise it's fairly logical. You must pay close attention to everything you're told though.

This game has virtually all the features I expect from an adventure, and shows how powerful The Quill utilities can be. Because there's so much, the game is in three parts; you can move freely between them reloading SAVED positions. A little inconvenient but far better than having less game.

There are a few bugs and the game is no masterpiece, but despite its faults The Boggit is an extremely enjoyable and professional adventure.

MONSTER HITS

ZX COMPUTING

FULL CREDIT FACILITIES FOR ACCESS AND BARCLAYCARD.

Lineage: 40p per word.

Semi display: £9.00 per single column centimetre. Ring for information on series bookings/discounts (minimum 2.5cm).

All advertisements in this section must be prepaid. Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01 - 437 0699

Send your requirements to:
DUNCAN NEILSON
ASP Ltd., 1 Golden Square, London W1

SERVICES

CASSETTE COPIES

50 C 10's FOR £40 inc. P&P load tested.
Send your cassette master c.w.o. to:
SOUNDS GOOD
'Clear Croft' Bottom Lane, Seer Green, Bucks HP9 2UH
(04946) 5900

UTILITIES

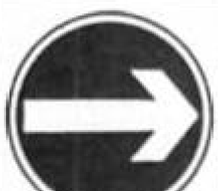
BRICKWORKER

A program for the DIY person to calculate the qty's of materials for small brickwork & concreting projects.
* *ILLUSTRATIONS OF BONDING * FOUNDATIONS * BRICKWORK CALCULATOR * MORTAR MIXES * POINTING * DECORATIVE FINISHES*
FOR THE 48K SPECTRUM £3.95 inc p&p
Send cheque to:
KONKRETE SOFTWARE
6 Willowcourt Ave, Kenton, Harrow, Middx.

SOFTWARE

ZX81/16K I CHING. Draws first three heragrams, prints results. Cheque £2.99 Eric Rosenbaum PO Box 2155 Jenkintown Pa, 19046 USA. Postpaid. Source Listing.

QL/SPECTRUM UTILITIES! Wide range of business/practical programs available. SAE/IRC for details (state Micro). S.D. Microsystems (ZX) PO Box 24, Hitchin, Herts.



TAKE A STEP IN THE RIGHT DIRECTION

PHONE DUNCAN NEILSON ON 01-437 0699 FOR ADVERTISING

REPAIRS

SINCLAIR REPAIRS

SPECTRUM/PLUS	£12.00
KEYBOARD	£8.50
INTERFACE/MICRODRIVE	£17.00 each

All prices are fully inclusive of p&p and VAT. Send Cheque with computer only unless power supply suspected. Callers Welcome.

I.T. WESTERN ELECTRONICS
Unit F2A & F3
Avonside Enterprise Park
Newbroughton Road,
Melksham, Wilts
Tele: (0225) 705017

SPECTRUM REPAIRS

We expertly repair Spectrums for £17 inclusive of parts and labour and return postage.

Spectrums upgraded to 48K for £34 inc. P&P. All keyboard faults cost only £12.

R. A. Electronics,
133 London Road South,
Lowestoft, Suffolk.
Tel: (0502) 66289.

SINCLAIR SERVICE CENTRE
- Fully Guaranteed Fast Repair Service
- Approx. 24hr Turnaround
- £15 av cost or send machine with £1.95 (return p&p) for free estimate.
- Personal callers welcome.

QUANTUM
33 CITY ARCADE, COVENTRY CV1 3HX
Tel: (0203) 24632

INTERCONTINENTAL SPECTRUM REPAIRS

Spectrum/Plus	£19.95
Spectrum Keyboard Faults	£12.00
Reset Sw. fitted with repair	£3.00
Upgrade 16K to 48K	£25.00
Interface 1	£17.50
Spectrum + Dustcover	£1.95
QL Dustcover	£2.95

Prices includes parts, P&P
Send Unit & Cheque/PO to:
P.E.A.K. ELECTRONICS
222 York Road, Hartlepool, Cleveland TS26 9QN.
Tel: (0429) 233199
Overseas please add £3.50 p&p

SPECTRUM REPAIRS

£14.95 inclusive of labour parts and **p&p.** Fast, reliable service by qualified engs., average repair **24hrs. 3 months guarantee.** For help or advice ring:
H. S. Computer Services, Unit 2, The Orchard, Warton, Preston, Lancashire PR4 1BE. Tel: (0772) 632686.

SINCLAIR COMPUTER REPAIRS

All machines despatched within 72 hours of receipt.
Spectrum **£17.00**. ZX-81 **£12.50**
Spectrum 16 to 48K upgrade **£25.00**
(inc. parts, VAT and return postage)
For a fast and reliable service, send machine, brief description of fault and cheque or postal order to the experts:
Capital Computer Services,
Unit K2, Cardiff Workshops, Lewis Rd, East Moors, Cardiff CF1 5EG.
Tel: (0222) 461801

SCOTLAND'S No 1

For home and personal computer repairs
Specially fast Spectrum service!
- *Same day for most faults*
- *1 hour if delivered personally*
- *Open 6 days a week*
- *Free estimates*
- *Upgrades, Membranes and P.S.U.'s*
- *3 mth. warranty on work done*

Also BBC/CBN/ORIC and PERIPHERALS

MICRO-SERV
95 Deerdykes View
Westfield Industrial Area,
Cumbernauld G68 9HN
Scotland
Tel: Cumbernauld (02367) 37110
Trade, schools and club discount given.

WOODHOUSE ELECTRICIAL SERVICES.

2 Station Rd, Woodhouse, Sheffield S13 7QL

Sinclair/Spectrum repairs from **£19.95** (return p&p inclusive)
* Only genuine Sinclair parts used
* Quick 48hr turn around
* Fast reliable service
* 3 month guarantee on work done
* All callers welcome 9-5.30 Monday to Saturday.

Tel: (0742) 695319.

ACCESSORIES

SPECTRUM RESET SWITCH - Prevents damage to your computer when power plug pulled in and out. **Only £3.49 inclusive.** Cheque or P.O. to Central Electronics, St Andrews St, Greenock PA15 1HG. Other accessories. Write for free list.

FOR SALE

STARCRASH. More than just an adventure. "The text really is some of the best yet seen on the Spectrum". Mindplay September 86. "Mandatory fare for Spectrum owners". Tony Bridge (PCW) June 19th. £4.95 inc. P&P from: Starcrash Software Systems. 1 Walsingham Avenue, Manchester M20 8WA.

PHONE NOW ON 01-437-0626 FOR ADVERTISING DETAILS.

ACCESSORIES

RGB OUTPUT ?
COMPOSITE VIDEO ?
GREEN SCREEN ?
TTL · ANALOGUE ?
COMP SYNC ?
BRIGHT ?

adapt electronics

CONFUSING ISN'T IT?

Talk of the problem solvers. We have the experience and expertise to advise on all aspects of computer display problems. eg:
★ **RGB OUTPUT MODULE** for Spectrum 16 or 48K. Much acclaimed — gives monitor quality pictures — no dot crawl. £36.95
★ **NEW ★ BRIGHT BOX** — Allows full use of the bright facility on analogue input monitors and monitor-TV's to produce full colour palette. Only for Spectrum 128. .... £29.95

Add P&P £1 (UK), £2 (EEC), £4 (Others)
Send SAE for our Free Monitor Fact Sheet

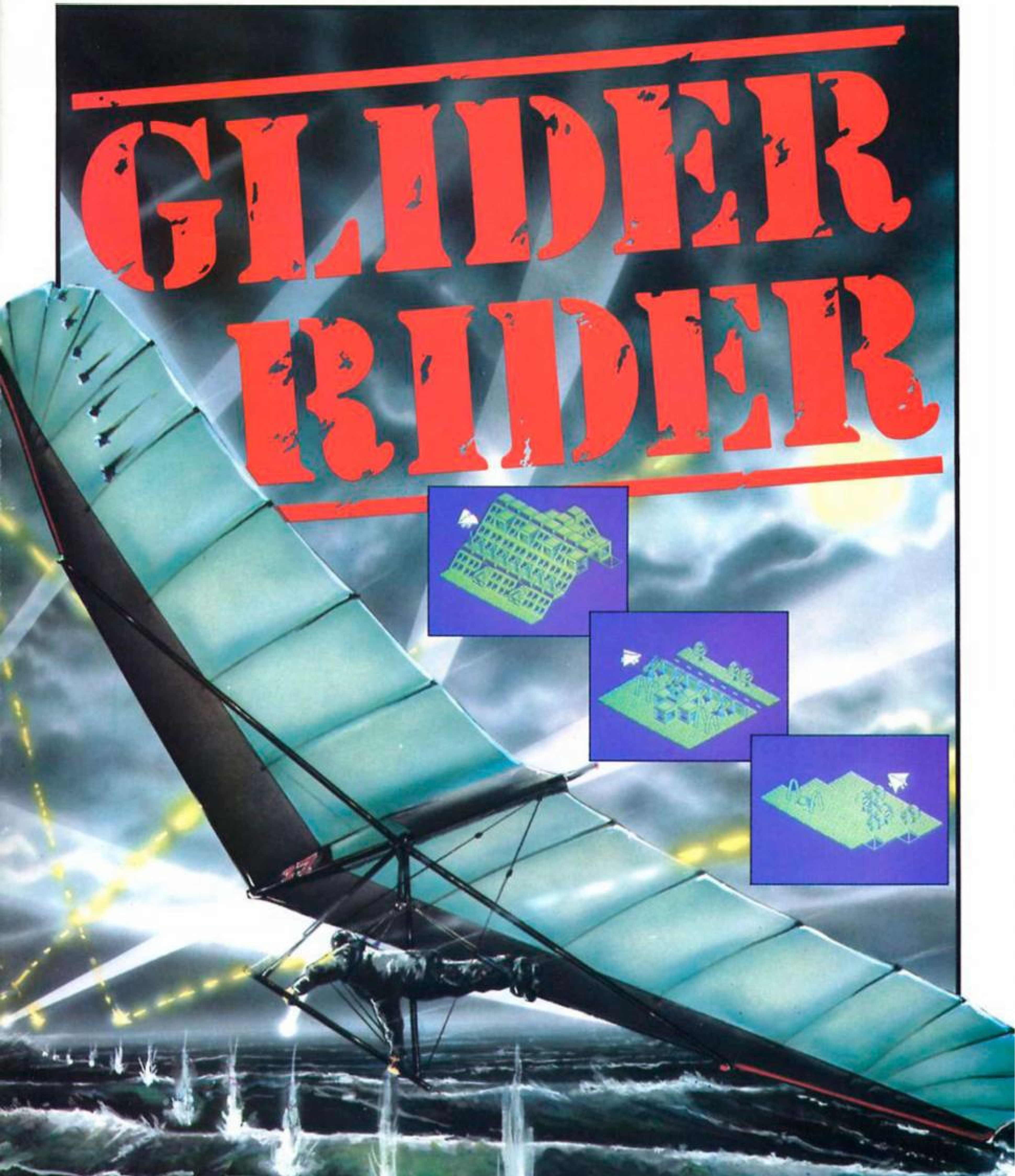
20 STARLING CLOSE, BUCKHURST HILL, ESSEX IG9 5TN. Tel: 01-504 2848

WARNING NOTICE

Advertisements placed in this magazine are to be in strict compliance with our standard conditions (copies of which conditions are available on request) and on the clear understanding that the advertiser warrants that his advertisement(s) does not infringe any copyright or condition of sale of any interested party in the advertised product.

Further, the advertiser indemnifies the proprietors of this magazine in respect of costs, damages, or any other claims brought against them as a result of legal action arising from the publication of the advertisement.

Any breach of these terms or the said conditions may result in prosecution of the advertiser by the proprietors.


GLIDER
RIDER


THE
GREAT
ESCAPE
ocean
Spectrum 7.95 each
Commodore 8.95 each Amstrad
Ocean Software Limited
6 Central Street Manchester M2 5NS.
Telephone: 061 832 6633 Telex: 669977